Prediction of Minimum Miscibility Pressure using Group Method of Data Handling

by

Mohd Amierul Azhar

13570

Dissertation submitted in partial fulfilment of

the requirements for the

Bachelor of Engineering (Hons)

(Petroleum Engineering)

MAY 2014

Universiti Teknologi PETRONAS

Bandar Seri Iskandar

31750 Tronoh

Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Prediction of Minimum Miscibility Pressure using Group Method of Data Handling

by

Mohd Amierul Azhar

13570

A project dissertation submitted to the
Petroleum Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirements for the
BACHELOR OF ENGINEERING (Hons)
(PETROLEUM)

Approved by,

_____
(Dr. Mohammed Abdalla Ayoub Mohammed)

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
May 2014

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own expect as specified in the references and acknowledges, and that the original work contained herein have not been undertaken or done unspecified sources or persons.

_____
(Mohd Amierul Azhar)

ABSTRACT

Enhanced oil recovery benefits the oil and gas industry by increasing the oil production that had been declined in particular reservoir. One of the methods is by miscible gas injection. In order to inject the gas into the reservoir and achieve miscibility, the determination of minimum miscibility pressure (MMP) is important. This report is regarding the study of MMP by using group method of data handling (GMDH). There are many ways to determine MMP. However, selection of GMDH as a method had been done because most of the experimental methods take a lengthy time to be completed. Although through previous studies, a lot of models had been produced, still, the industry in need of researches that generate correlations that are better in accuracy. Therefore, this study aimed to model MMP with GMDH, using common correlated parameters, and made comparison with the previous model. In this report also, researches been done by gathering data required by doing comprehensive readings on particular topics related to the projects especially regarding the concepts of MMP and GMDH. Only then the model could be made and optimized ultimately producing an accurate and reliable correlations.

ACKNOWLEDGEMENT

Upon completion of this project, the author have gained a lot of knowledge and experience. However, there is no achievement that did not come with struggles and problems. Therefore, the author realized, without the presence of several key people, there are possibilities that he would have difficulties to succeed in this project. Therefore, first and foremost, I would like to thank Madam Noor Ilyana Ismail, coordinator for Final Year Project I, Mr. Muhammad Aslam Md Yusof, Dr. Mohana Sundaram Muthuvalu and AP Dr. Subhash Kamal, coordinators for Final Year Project II, for their hard works, managing and guiding the author specifically in term of project management; holding seminars and assigning datelines. Special thanks to Universiti Teknologi PETRONAS Petroleum Engineering student Batch May 2009, for knowledge sharing throughout the project period, in person or by online method. Highest gratitude would like to be given to Dr. Mohammed Abdalla Ayoub Mohammed, Final Year Project Supervisor, for the supervision and direct instruction on working out the project until the end. Their contribution had been enormous in order for the author to gain inputs and successfully completed the project. The author hope that every person that gives contribution, directly or indirectly, during this internship period will gain blessing and mercy from Him, the Most Merciful. With that again, thank you.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

MMP	Minimum miscibility pressure	GMDH	Group Method Data Handling
EOR	Enhanced Oil Recovery	OOIP	Original Oil in Place
RBA	Rising Bubble Apparatus	VIT	Vanishing Interfacial Tension
PVT	Pressure, Volume and Temperature	ANN	Artificial Neural Network
C_{2-6}	Ethane through Hexane	CO_2	Carbon Dioxide
H_2S	Hydrogen Sulfide	C_1	Methane
C_{2+}	Ethane plus	C_{7+}	Heptane plus
VB	Flank Wear	ISO	International Standard Organization
PSE	Predicted Square Error	APE	Average percent relative error
AAPE	Average absolute percent relative error	E_{min}	Minimum absolute percent relative error
E_{max}	Maximum absolute percent relative error	RMSE	Root mean square error
SD	Standard deviation	R	Correlation coefficient
FYP	Final Year Project		

CHAPTER 1
INTRODUCTION

1.1 Background

EOR has been a main topic for the last decades. With declination of hydrocarbon production as a factor, EOR gained attentions due to its ability to improve the productivity of a reservoir. One of the most effective method in EOR is the miscible flooding. One of the condition that affects the selection of miscible flooding is the MMP. At reservoir temperature, a condition where the injected gas mixed with reservoir oil, the lowest pressure at that particular point is regarded as MMP. Therefore, a decrease in oil recovery can be observed if the reservoir pressure drop below MMP, resulting no miscible displacement can be achieved. Hence, considerable interest had been given to the industry to predict the minimum miscibility pressure of certain reservoir. Production of residual oil left in the reservoir will mostly depending on the EOR method. Consequently, obtaining the minimum miscibility pressure will be vital in determining preferable EOR method to be applied to the respective reservoir, increasing the production of oil and gas.

There are a lot of technique can be done in order to predict the minimum miscibility pressure. Ranging from experimental to analytical approaches, many studies had been done in regards of this topic. GMDH has been proposed in this study to predict the MMP.

GMDH is a self-organizing network where the structure of the network is not defined by the users. GMDH works with the concept to link several input parameters to one output. GMDH consume less in term of time usage as the stopping criterion is specified by the users. Although the effectiveness of GMDH in oil and gas sector is still in doubt due to little references that can be found relating to this good modelling method, this approach had been selected for this study due to the fact of its benefits in contrast to other method.

1.2 Problem Statement

Few problems have been identified related to the study of minimum miscibility pressure determination using group method data handling:

i) Tedious experimental procedure.

Determining the minimum miscibility pressure by using the experimental methods such as slim-tube test would be a better solution. However, such procedure would take more time in order for the test to be completed. This will exceed the time frame of the study. Taking an analytical approach will save more time.

ii) Existing models and correlations are complex and lack in accuracy.

Several assumptions and considerations had been taken to reach the current models and correlations. There are possibilities that those assumptions made and particular considerations are actually important to the prediction of minimum miscibility pressure. Thus, every details need to be taken into consideration before arriving at a simplified model yet with a high accuracy.

iii) More general and universal correlation need to be introduced to the petroleum industry.

By applying existed correlations, at certain conditions, the percentage error exceed 100%. This shows that the correlations are bit off and over fit the experimental data used during the development of the correlation. Hence, a correlation that can cover a wider scope is needed by the industry in order to predict the minimum miscibility pressure.

1.3 Objectives and Scope of Study

The objectives of this study are as follow:

i) To model the MMP using GMDH approach with the most common correlated input parameters.

GMDH method was selected to model the MMP. Besides having other experimental and analytical methods, a simpler, more applicable and reliable correlation will be generated to ease the prediction of MMP. There are several parameters involved in determination of MMP. However, most common input parameters, to be used in GMDH, will be determined in order to have a general correlation. General here can be defined as those parameters that frequently been used to predict the MMP.

ii) To validate and test the model performance using actual data.

Validation and testing is important once a model is being developed to ensure that the correlation is in working condition. On top of that, the usage of actual data preserve the originality of the results produced. Hence, by validation and testing, the output and accuracy of the correlation can be assured to be used generally in the industry.

iii) To compare the model performance against the most common correlations adopted by the industry.

There were several correlations that can be found in the industry regarding the prediction of MMP. However, comparison will be made between model produced in this study and the others to see the increment in accuracy and quality of the model generated using GMDH. A simple correlation is to be determined that does not fit the training with high accuracy from this study. A dimensionless model is also expected to ease the usage of the correlation.

Within the stipulated time frame, the study must be done accordingly and covering very specific scope to achieve all the objectives mentioned before.

i) Literature survey

An in depth study from any sources of literature relating to project's topic need to be done. Comprehension of the topic need to be obtained, focusing on 3 points as follow:

a) Concepts of MMP

A detailed study need to be done on methods of MMP prediction where several methods and results will be discussed. Previous correlations obtained from the any available and trusted sources of reference will be used to search for parameters and factors that give major impact on the minimum miscibility pressure. These parameters and factors will be used in the validation and testing of the models produced and to compare with the previous correlations. Concurrently, evaluation studies will be done on the accuracy of the parameters and existing correlations to help in the MMP models generation. This is due to considerations taken of parameters that might have vital value must not being missed out during the modelling construction.

b) Overview of GMDH

Readings on modelling, basic concept of GMDH will be done to understand how GMDH operates and how it can be implemented in prediction of MMP.

c) Application of GMDH in petroleum industry

Researches are made in order to have a clear view on how GMDH has been affecting the oil and gas industry.

ii) Model simulation

By utilizing the GMDH approach, a model development will be conducted with the inclusion of all parameters and factors identified which affect the prediction o minimum miscibility pressure. This development phase will require data that will be provided later on.

iii) Correlation validation

Prior to the completion of model in development phase, training, validation process and testing will be conducted based on the data collected. Therefore, the validity and accuracy of the correlation can be assured. This process will be repeated until the desired accuracy of the model is achieved.

CHAPTER 2
LITERATURE REVIEW

This chapter summaries all comprehensive study made on the selected literature regarding minimum miscibility pressure. It can be divided into three (3) subtopics which include

2.1 Concept of MMP

EOR or enhanced oil recovery is a mechanism to further increase the production of oil. Being regarded as the tertiary oil recovery phase, some highly cost equipment are deployed as part of EOR in order to produce up to 75 percent of OOIP. Three (3) EOR's main types are always being discussed. These are chemical injection, thermal recovery and gas injection. Thermal injection involves an introduction of heat into the reservoir. This will eventually decrease the viscosity of the oil, usually used in a heavy oil reservoir, and ease the production of the oil. While chemical injection refers to those methods where chemical formulations are injected into the reservoirs to help in oil production increment. For example, alkaline, surfactant and/or polymer are being used with each of them will bring different results, altering particular reservoirs parameters. Last but not least, where this study is mostly about is the gas injection. As the term goes by, gases are being used in EOR to improve oil production. In the secondary oil recovery phase, gas injection is also deployed. However, the difference between both gas injections is that the one in the secondary recovery is a mere injection of particular gas to displace the oil to the surface while in tertiary recovery miscible gas injection occurs. MMP plays a role in determining the miscibility of the reservoir fluids.

Miscible gas injection is a process of injecting gas into the reservoir to increase the production of oil. Gases being used include carbon dioxide (CO_2), flue gas, nitrogen gas and lean gas. The injected gas will be selected based on the analysis made on the reservoir characteristics. There are two (2) miscible gas injection mechanisms, first-contact miscibility and multiple contact miscibility. First-contact miscibility can be explained as when there are two fluids that mixed with each other upon the initial contact in contrast to multiple contact miscibility where exchange of fluids components happens

before the fluids mixed together. Multiple contact miscibility also consists of condensing drive and vaporizing drive. These drives also affected by particular reservoir characteristics. In condensing drive mechanism, intermediate components will enter the oil phase from the gas phase while it is the other way round for light and intermediate components in vaporizing drive mechanism. Anyway, MMP affects any of the mechanism above.

A condition where the oil and injected gas mixed in the reservoir is called MMP. Usually, there are forces that prevent two immiscible fluids from mixing called the interfacial tension. However, as the MMP approaches, interfacial tension will diminish and the absence of interface between the two fluids causes miscibility.

Being a very important parameter in EOR method's selection, a lot of studies been done in order to determine MMP and studies done showing there are several methods to determine it, be it by experiments or numerical approaches. It was stated that MMP can be determined experimentally with slim tube test, RBA test and VIT technique. (Chen, G., Wang, X., Liang, Z., Gao, R., Sema, T., Luo, P., Zeng, F., and Tontiwachwuthikul, P., 2013) [1]. According to Maklavani et al. [2], slim tube test is being adopted as the most typical method to measure MMP in the industry. However, they reinstated that the method take a lot of time and resorted to the use of a slim tube compositional simulator. In their study, the model consists of as following: rectangular reservoir with 20 meter in length and 0.000025m2 in cross-sectional area, hydrocarbon gas injected at 10 cc/h, porosity at 20%, permeability at 1000 md and 500 blocks. Initially, the model was saturated above the bubble point pressure and at reservoir temperature with oil. Peng-Robinson equation of state parameters was tuned and brought into the simulator in order to simulate the reservoir fluid. Before MMP was determined with this method, known oil MMP and PVT data was used to validate and calibrate the simulator.

Other than that, Glaso [3] also mentioned that miscibility conditions can be determined with experimentation. He added that it is also possible to predict MMP from ternary diagrams. Calculations made from gas and liquid compositions of a reservoir

oil/injection-gas mixture is the key to this. Glaso also stressed that it is hard to get on experimental gas and liquid equilibrium data besides taking a long time to determine one, especially near the plait point.

One of the popular numerical method is ANN modeling approach. By user-definition of significant input parameters and ratio of network's layer, used to establish the connection between multiple inputs and an output, ANN modeling succeed in giving new predictions with higher accuracy with just a simple model. Therefore, based on the application of this method, it was proven that ANN modelling gives better predictions with higher accuracy and lower complexity compared to the existing correlations or models. However, ANN has its limitations. ANN takes a lot of time to reach finalized model. This is because trial and error need to be done to set various algorithms for the initial solution which affects the results produced. To estimate the structure of the model in ANN, the used need to decide the quantity of layers and transfer functions of nodes. This process can be very subjective and tedious. These two limitations hinder ANN potential to give a good prediction to minimum miscibility pressure.

Several previous correlations identified can be used to determine MMP:

i) Maklavani, A. M., et al. [2] Correlation

This correlation is a result of regressed measured MMP data determined by experiments and also from compositional slim tube simulator. It includes the effect of temperature, oil composition and injected gas composition.

$$MMP = 43.664 - 4.542\alpha + 0.689\alpha^2 - 0.132\beta$$

$$\alpha = \frac{X_{C_{2-6}}^{1.72785} \times X_{C_1}^{0.1}}{(1.8T + 32)^{0.5} \times M_{C_{7+}}}$$

$$\beta = Y_{C_{2+}}^{\left(1.064 + 0.00686 M_{C_{2+}}\right)}$$

Parameter

MMP: minimum miscibility pressure (MPa)

$X_{C_{2-6}}$: intermediate composition in the oil containing $C_{2\text{-}6}$, CO_2 and H_2S in mole %

X_{C_1}: amount of methane in the oil (%)

T: temperature (°C)

$M_{C_{7+}}$: molecular weight of C_{7+} (g/mol)

$Y_{C_{2+}}$: mole percent of C_{2+} composition in injected gas (%)

$M_{C_{2+}}$: Molecular weight of C_{2+} in injected gas

ii) Sebastian and Lawrence [4] Correlation

This correlation includes effects of oil composition and reservoir temperature to determine the MMP of nitrogen.

$$N_2\ MMP = 4603\ - 3283\left[\frac{(CL)(T)}{MW}\right] + 4.776\left[\frac{(CL^2)(T^2)}{MW}\right] - 4.008\left[\frac{(CI)(T^2)}{MW}\right] + 2.05\ (MW) + 7.541T$$

Parameter

N_2 MMP: nitrogen minimum miscibility pressure

CL: mole fraction of methane in the oil

CI: mole fraction of intermediates ($C_{2\text{-}6}$ and CO_2)

T: reservoir temperature (°R)

MW: molecular weight of C_{7+}

iii) Firoozabadi and Aziz [5] Correlation

This correlation utilized experimental data and simulation runs to develop a simple correlation to predict the MMP.

$$p_m = 9433 - 188 \times 10^3\left(\frac{C_{C_2} - C_{C_5}}{\left(M_{C_{7+}}\right)\left(T^{0.25}\right)}\right) + 1430\ \times 10^3\left(\frac{C_{C_2} - C_{C_5}}{\left(M_{C_{7+}}\right)\left(T^{0.25}\right)}\right)^2$$

Parameter

P_m: MMP (psia)

T: temperature (°F)

$C_{C_2} - C_{C_5}$: concentration of intermediates, (mol %)

MC_{7+}: molecular weight of heptane plus

2.2 Overview of GMDH

Fujimoto and Nakabayashi [6] stated that GMDH was brought into the scene in 1966 by Ivankhenko. Up until now, it has been known that GMDH is used in complex system modeling as an inductive learning algorithm for complex systems modeling.

GMDH is an inductive self-organizing methods that was modified further to solve complex problems. It is built from complex, non-linear relationships, compacted into networks of mathematical functions that execute output faster than its original forms. A problem is divided into nodes predefined by the users. Then, those simpler problems are solved by using advanced regression techniques. That is what self-organizing is all about. Astakhov and Galitsky [7], in their study, reiterate to simplify that, GMDH works by establishing a general network, as shown in Kolmogorov-Gobor polynomial below. It requires inputs and producing output which is solution to the problem.

$$y = b_0 + \sum_{i=1}^{M} b_i x_i + \sum_{i=1}^{M}\sum_{j=1}^{M} b_{ij} x_i x_j + \sum_{i=1}^{M}\sum_{j=1}^{M}\sum_{k=1}^{M} b_{ijk} x_i x_j x_k \quad (1)$$

The component of input can be either independent variable, functional forms or even finite difference terms. In addition to that, GMDH method simultaneously enables optimization of a model structure and produces a model that takes significant inputs for the output to depend on. In other words, GMDH excludes unimportant parameters and takes inputs that really affect the output, hence the model produced.

Many fields, including medical diagnostics, military system, economy system and even environment system took the application of GMDH and succeed in doing so. However, there is very few application of this method in oil and gas sector although no valid reasons identified. Nevertheless, GMDH has a very high potential to be applied in oil and gas industries. Due to those uncertainties encountered in exploration, production and transportation especially, GMDH might produce reliable solutions to be applied. Therefore, if GMDH were to be compared with ANN, as per mentioned earlier, it is

expected an improved correlation will be produced besides overcoming all the limitations possess by ANN.

2.3 GMDH and current Petroleum Industry

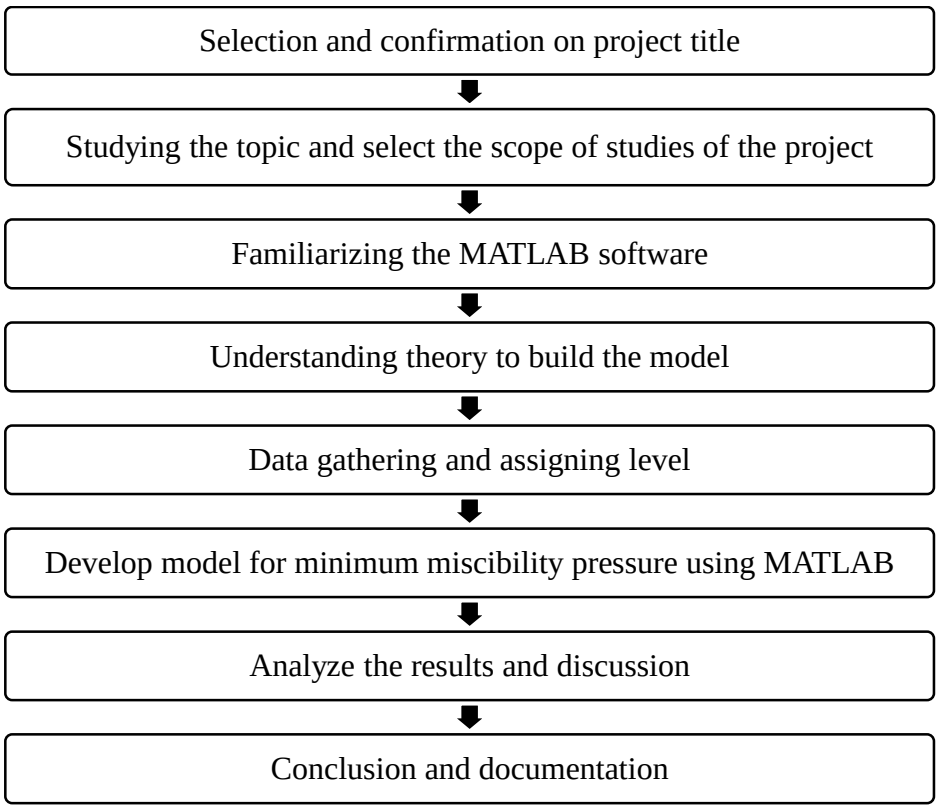
Semenov et al. [8] were developing the best mathematical model for Dolgan reservoir rock characteristics estimation. To conduct the study, all well logs that can be obtained was utilized. Among the methods being used in their study are linear regression, neural networks, and GMDH which end up of GMDH method having the best correlation obtained based on statistical criteria. It is able to optimize model coefficient for predetermined mathematical equation and select optimal model complexity. The result of porosity calculation with GMDH eventually help in the realization of a full-field geomodel of Dolgan formation. The geomodel is then implemented on model simulation and field development design to decide the recovery rate and either to drill horizontal or side track wells, respectively.

Lee et al. [9] used GMDH to determine drill life with various cutting conditions. Drill diameter, cutting speed, and feed rate have been selected as the parameters in this study. The drill life used in the abductive network is defined as the period of drilling time that the average flank wear, VB or maximum flank wear, VBmax, are equal to 0.3 mm or 0.6 mm respectively. This criterion is the recommended by the ISO for effective life of high-speed steel tools. From the experimental result, it was then proven that abductive network, obtained prediction error of less than 9%, can be effectively used to make prediction of this study. On top of that, PSE was utilized. PSE were used as benchmark with the means of synthesizing as accurate but less complex network as possible.

CHAPTER 3
METHODOLOGY

This chapter covers all the methodology that will be used, project activities and key milestones on this study. In general, simulation software will be greatly used in the development of the model by using GMDH approach. Further explanation on GMDH approach will be explained below.

3.1 Basic Layout of Planned Methodology



3.2 Project Planning

Towards the completion of this project, a proper planning need to be made. Therefore, it was vital for the author to identify the problems that need to be addressed as soon as the topic was confirmed. The formulation of problem was then conducted and three (3) problem statements were identified and understood. To solve the problems

mentioned, next the author came out with the objectives of this study. Three (3) objectives determined was actually to help in solving the problems statements, in a more focused manner.

In order for the planning to be commenced, the author need to take his resources into considerations. Resources in this case can be defined as inputs that will be needed with the author throughout the project's time frame which include materials, manpower, money and also time, in order to produce an output for this project. Determination of those particular aspects will help in determining this project magnitude. Since this was a simulation project, materials required do not need any usage of money and huge manpower. Only high specification computer was needed and the manpower was the author himself. As for the timing aspect, the duration and key milestones were decided so that the project will be completed on time.

The next step was to conduct a literature review. Works by previous authors were obtained from publications that can be accessible either at the library or from the internet. This was done in order to obtain better understanding regarding the project which include on MMP determination and the operation of GMDH.

3.3 MATLAB software

To complete the model, MATLAB was utilized. MATLAB is an advanced software used in numerical computation and programming. Data analysis, algorithms development and of course, to build a model can be done in MATLAB. Figure 1 and 2 below are the interface of MATLAB R2013a:

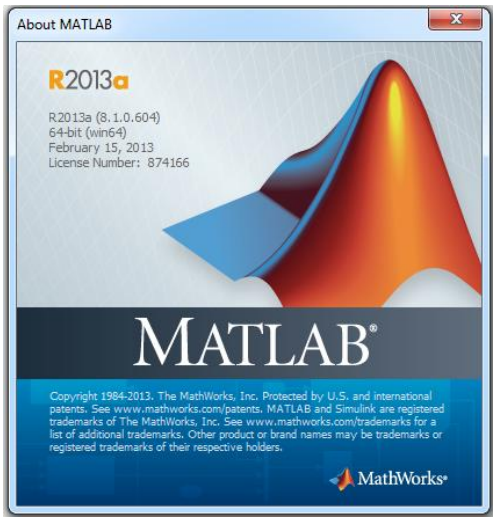


Figure 1: MATLAB Software

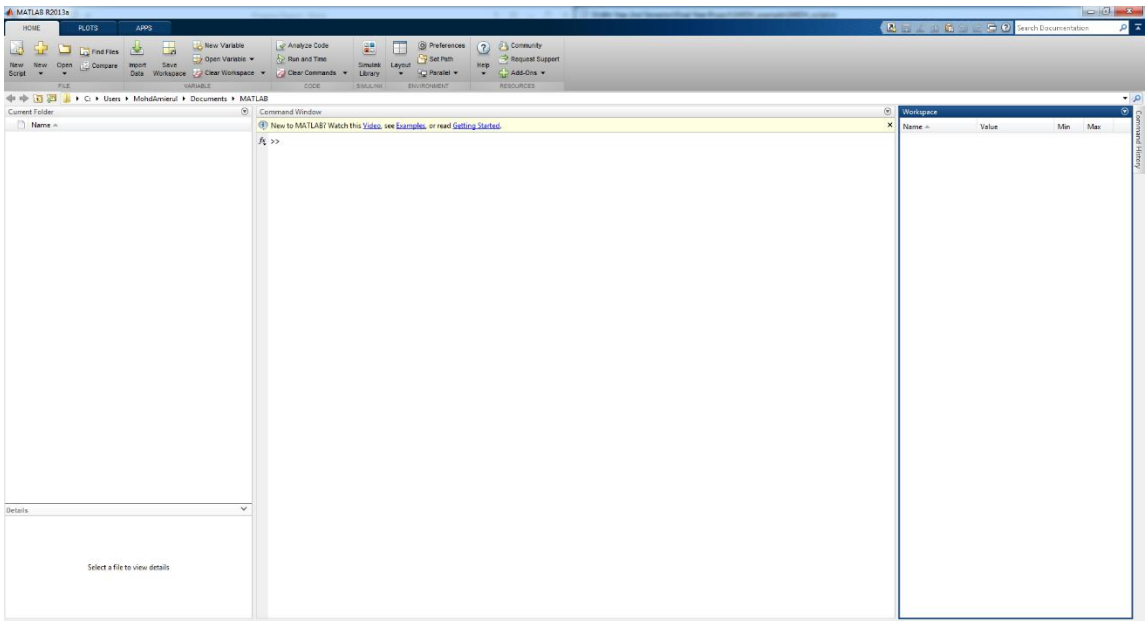


Figure 2: MATLAB Software Interface

Usually MATLAB is utilized in:

i) Modelling, making prototype and simulation

ii) Developing algorithms

iii) Analyzing data

iv) Producing scientific visualization and engineering graphics

v) Building graphical user interface for application

For this project, the author used a syntax in order to simulate the operation of GMDH. All the data was compiled in a Microsoft Excel file to be linked with the code. Eventually, a correlation was produced as a result of the process.

3.4 GMDH Modeling Approach

A study done on GMDH helped in understanding and developing a theory to build the model. Below is the basic technique of GMDH learning algorithm:

i) Identify sets of data containing an output, y and independent inputs namely $x_1, x_2, \ldots, x_M$. Divide them into a training set, a validation set and a testing set.

ii) Put in all M input variables data (Input Layer) and construct combinations for every two (2) variables for the second layer (Layer 2).

iii) By using training set, estimate the weights of all units ($b_0 \; until \; b_{ijk} \; in \; equation \; 1$) using stepwise regression method.

iv) Compute mean square error between y and prediction of each unit.

v) Arrange the unit by mean square error and discard poor units.

vi) Set the prediction of units in the first layer to new input variables for the next layer and build up a multi-layer structure by applying step (ii) to (iv).

vii) Stop adding layers and select the minimum mean square error unit in the highest layer as final model output when the mean square error become larger in the previous layer.

Figure 3 shows the basic structure of GMDH.

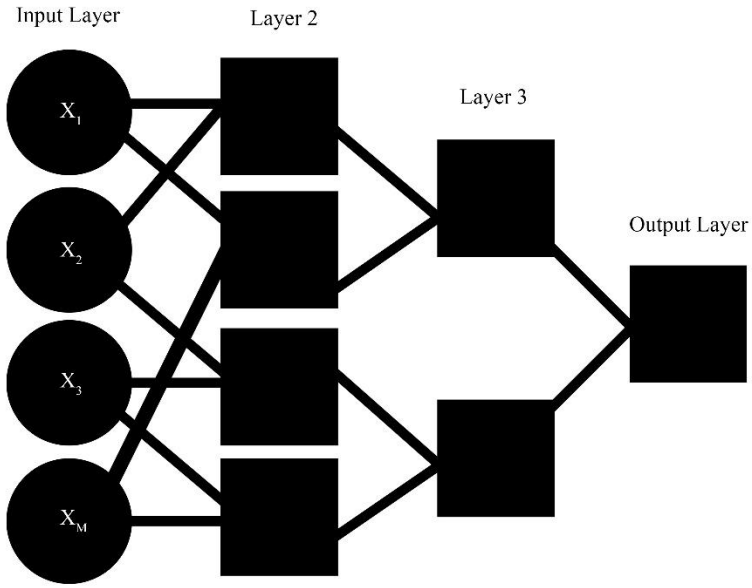


Figure 3: GMDH Structure

3.5 Data Acquisition and Processing

Data was needed in the process of building a model by using GMDH approach. Therefore, actual data of crucial parameters were started to be collected from trusted sources as per mentioned before. Parameters involved was selected from the studies done to obtain the correlations mentioned in literature review section. However, screening need to be done in order to select those parameters that are commonly obtained and used in the industry. Table 1 shows the screening of existing correlations done together with the parameters used. Finalized parameters selected for GMDH was based on the one that mostly used in the previous studies.

Table 1: Parameter Screening

Correlation	Parameters						
	Mole fraction of C_{2-6}	Mole fraction of C_1	Temperature	Molecular weight of C_{7+}	Mole fraction of C_{2+}	Molecular weight of C_{2+}	Bubblepoint Pressure
Maklavani, A. M., et al. [2]	✓	✓	✓	✓	✓	✓	
Sebastian and Lawrence [4]	✓	✓	✓	✓	✓		
Firoozabadi and Aziz [5]	✓	✓	✓	✓	✓		✓

Based on the screening process, it was decided that 5 parameters will be used as input variables for this study which are:

i) Reservoir temperature
ii) Molecular weight of C_{7+} in the oil
iii) Mole fraction of intermediates ($C_2 – C_6$) in the oil
iv) Mole fraction of C_1 in the oil
v) Mole fraction of C_{2+} in the injected gas

As the parameters were confirmed, data collection was began immediately. This is because a lot time was estimated in order to get the suitable data. 356 data sets used in this study is in **APPENDIX A**.

All data obtained need to be divided into portions. According to a ratio of 2:1:1, data partitioning was conducted into training, validation and testing data sets respectively. Training data took the highest fraction because this portion was used to develop the model in the first place. Validation and testing data shared the same portion as validation was needed for model optimization and testing data used to conform the final performance of the produced correlations.

3.6 Modeling Process Flow

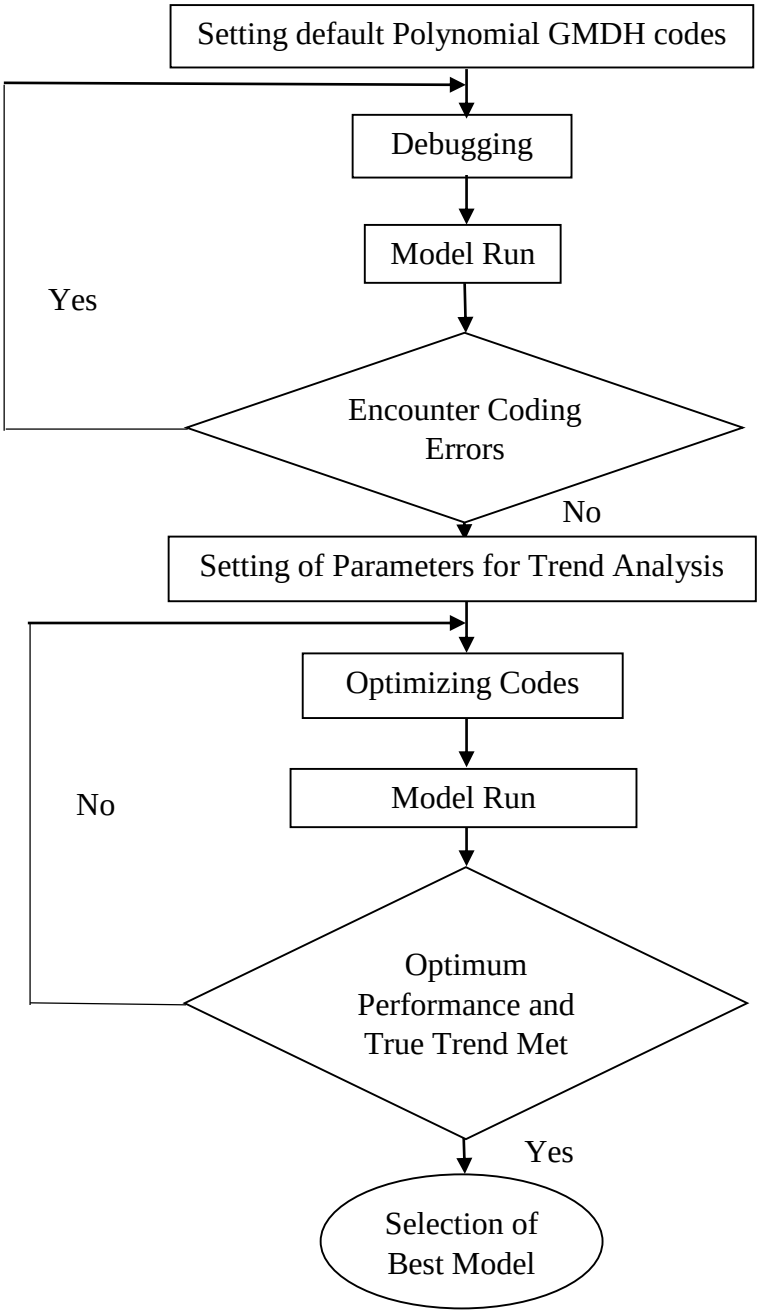


Figure 4: Modeling Process Flow

i) Setting default Polynomial GMDH codes

The GMDH coding was keyed in into the MATLAB software and modified suiting to the project objectives. This includes coding for the modeling and also statistical analysis.

ii) Debugging

The coding was debugged if any errors prompted by MATLAB. The errors prevent the program to run.

iii) Model Run

MATLAB was run. If errors were found, the coding need more debugging. If the run was error free, a model and statistical analysis graphs were produced.

iv) Setting of Parameters for Trend Analysis

The coding was then improved to have trend analysis. Specific parameter, depends on study done on them, was allowed to vary while the others were kept constant.

v) Model Run

MATLAB was run. If errors were found, the coding need more debugging. If the run was error free, a graph of trend analysis was produced to be compared with the real trend.

vi) Optimizing Codes

The coding were modified by trial and error to achieve optimum performance in terms of correlation coefficient and statistical analysis graph as well as obeying the original trend.

3.7 Types of Analysis

In order for the author to clarify the precision of the model generated, few analysis need to be done:

3.7.1 Group Error analysis

Group input parameters into a few ranges and determine the average absolute relative error for each model. Then, select which model that obtains the lowest average absolute relative error value. This is to show the reliability of the model with average absolute relative error is used to be the benchmark.

3.7.2 Statistical Error analysis

Calculate the value of average percent relative error, average absolute percent relative error, minimum and maximum absolute percent error, root mean square error, standard deviation or error and the model coefficients. This is to check the accuracy of the models.

i) APE (E_r)

$$E_r = \frac{1}{n}\sum_{i=1}^{n} E_i$$

$$E_i = \left[\frac{MMP_{measured} - MMP_{predicted}}{MMP_{measured}}\right] \times 100\% \, , where \; i = 1, 2, 3, 4, \ldots n$$

$$n = number \; of \; data \; set$$

ii) AAPE (E_a)

$$E_a = \frac{1}{n}\sum_{i=1}^{n} |E_i|$$

iii) E_{min}

$$E_{min} = min_{i+1}^{n} |E_i|$$

iv) E_{max}

$$E_{max} = max_{i+1}^{n}|E_i|$$

v) RMSE

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} {E_i}^2}$$

vi) SD

$$SD = \sqrt{\sum_{i=1}^{n} \frac{\left[\bar{x}_{i_{errors}} - x_{i_{errors}}\right]^2}{n-1}}$$

$$where\ x_{i_{errors}} = MMP_{measured} - MMP_{predicted}$$

vii) R

$$R = \sqrt{1 - \frac{\sum_{i=1}^{n}\left[MMP_{measured} - MMP_{predicted}\right]}{\sum_{i=1}^{n}\left[MMP_{measured} - \overline{\Delta MMP}\right]}}$$

$$where\ \overline{\Delta MMP} = \frac{1}{n}\sum_{i=1}^{N}[\Delta \mathrm{MMP}_{measured}]$$

3.7.3 Graphical Error analysis

Use cross-plots, error distribution and residual analysis to analyze the performance and accuracy of the model.

i) Cross-plots

To compare the predicted values and measured values, plot both values with a 45° straight line made to signify a perfect correlation for every set of data. The model said to be most accurate when the slope of the data is closest as the 45° straight line.

ii) Error distribution

To see either the errors are normally distributed, histograms are generated for each set of data. A good indicator is if the models have skewed distribution curve and the mean around the 0% and the standard deviation is equal to 1.0.

iii) Residual analysis

To check for any deficiencies on the model, error trends are established by plotting relative difference of the measured and predicted values around zero line.

3.7.4 Trend analysis

Keep the value of required parameter to be varied while the other parameters constant. A plot between the input parameter and minimum miscibility pressure is then generated and analyzed. This is to validate either the model generated obey the real behavior of the input parameter or not.

Other than those, the testing data sets were also used for comparison between the correlation generated and previous correlations.

3.8 Reporting

At this stage, the project was concluded as final report preparation went on. Prediction of MMP by utilizing GMDH was completed.

3.9 Gantt Chart and Key Milestones

Table 2: FYP 1 Gantt Chart

No	Detail/Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Selection of Project Topic														
2	Preliminary Research Work														
3	Submission of Extended Proposal														
4	Proposal Defence														
5	Project Work Continues														
6	Submission of Interim Draft Report														
7	Submission of Interim Report														

Table 3: FYP 2 Gantt Chart

No	Detail/Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	Project Work Continues																
2	Submission of Progress Report																
3	Project Work Continues																
4	Pre-SEDEX																
5	Submission of Draft Report																
6	Submission of Dissertation (Soft Bound)																
7	Submission of Technical Paper																
8	Viva																
9	Submission of Project Dissertation (Hard Bound)																

Process

Key Milestones

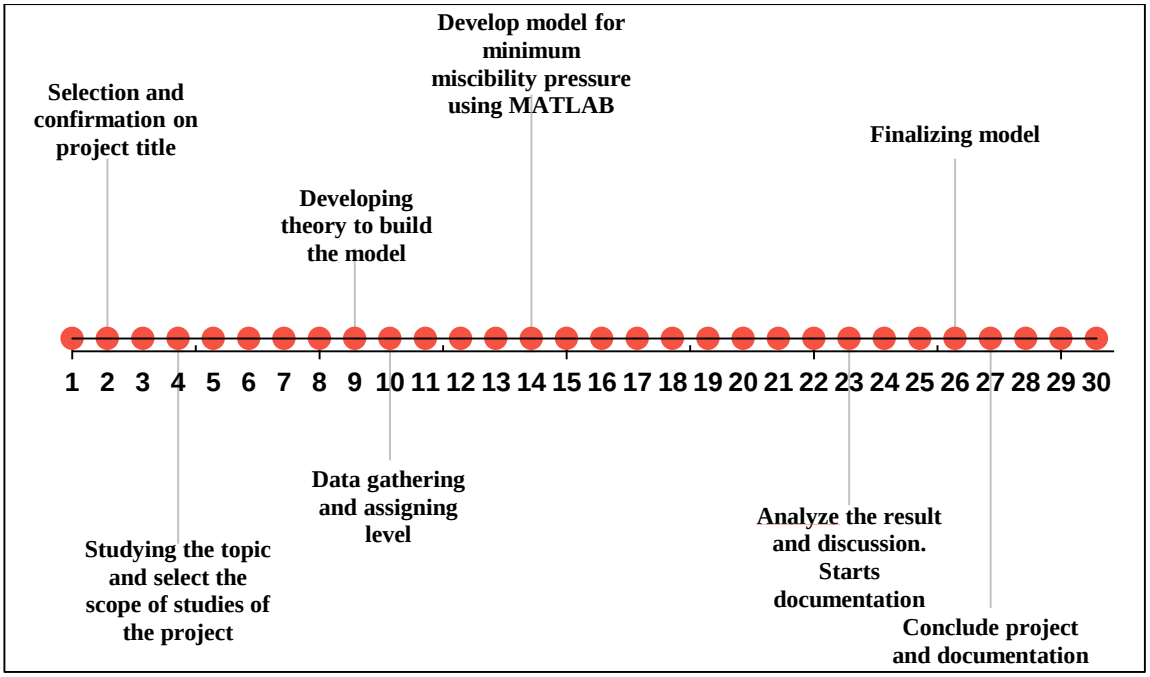


Figure 5: FYP Planning

CHAPTER 4
RESULTS AND DISCUSSION

4.1 Introduction

In this chapter, the correlation produced by GMDH will be analyzed by statistical and graphical methods. Apart from that, comparison with the previous correlations will be done and trend analysis will wrap up the analysis, ensuring the model is following the same trend.

4.2 MMP Model by GMDH

The model produced is made up of two (2) layers and ten (10) neurons were tried in the first layer which in the end with only one (1) neuron. As default, the second layer has the output, MMP, as the only neuron. The model produced show significant effect of mole fraction of intermediates (C_2 – C_6) in the oil, mole fraction of C_1 in the oil and molecular weight of C_{2+} in the injected gas towards the prediction of MMP. The model is as follows:

Layer 1

Number of neuron: 1

```
x6 = 1.04e+04 - 88.9*x5 - 841*x3 +7.19*x3*x5 - 3.14*x5*x5 +
44.2*x3*x3 - 0.0159*x3*x5*x5 - 0.0996*x3*x3*x5 + 0.0332*x5*x5*x5
- 0.71*x3*x3*x3
```

x3 = mole fraction of intermediates (C_2 – C_6)

x5 = mole fraction of C_{2+}

Layer 2

Number of neuron: 1

```
y = 2.87e+04 - 15.2*x6 - 533*x4 + 0.273*x4*x6 + 0.00274*x6*x6 - 2.69*x4*x4 - 2.58e-05*x4*x6*x6 - 0.000731*x4*x4*x6 - 1.43e-07*x6*x6*x6 + 0.0707*x4*x4*x4
```

y = MMP

x4 = mole fraction of C_1

4.3 Group Error Analysis

To compare the reliability of the new mode compared to the previous correlations, this analysis was completed with E_a used as the benchmark. The previous selected parameters by GMDH was selected and their values were grouped in several ranges and plotted.

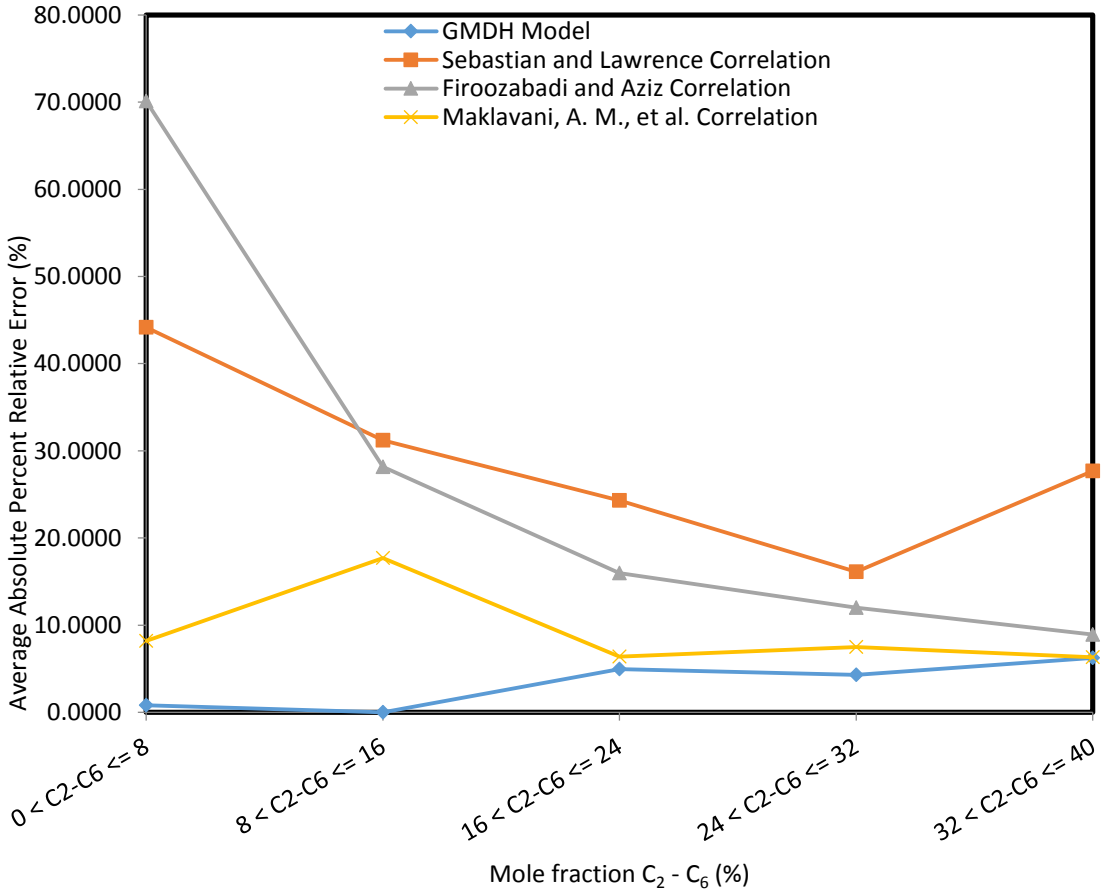


Figure 6: Statistical Accuracy for MMP with Different Mole Fraction of C_2 – C_6 Range

Different mole fraction of C_2 – C_6 range is shown in Figure 6. Throughout the ranges, GMDH recorded the lowest AAPE with Maklavani, A. M., et al. [2] correlation is the nearest to rival GMDH. This signify the stability of GMDH model compared to the other correlations as the AAPE for all mole fraction of C_2 – C_6 ranges does not exceed 10%.

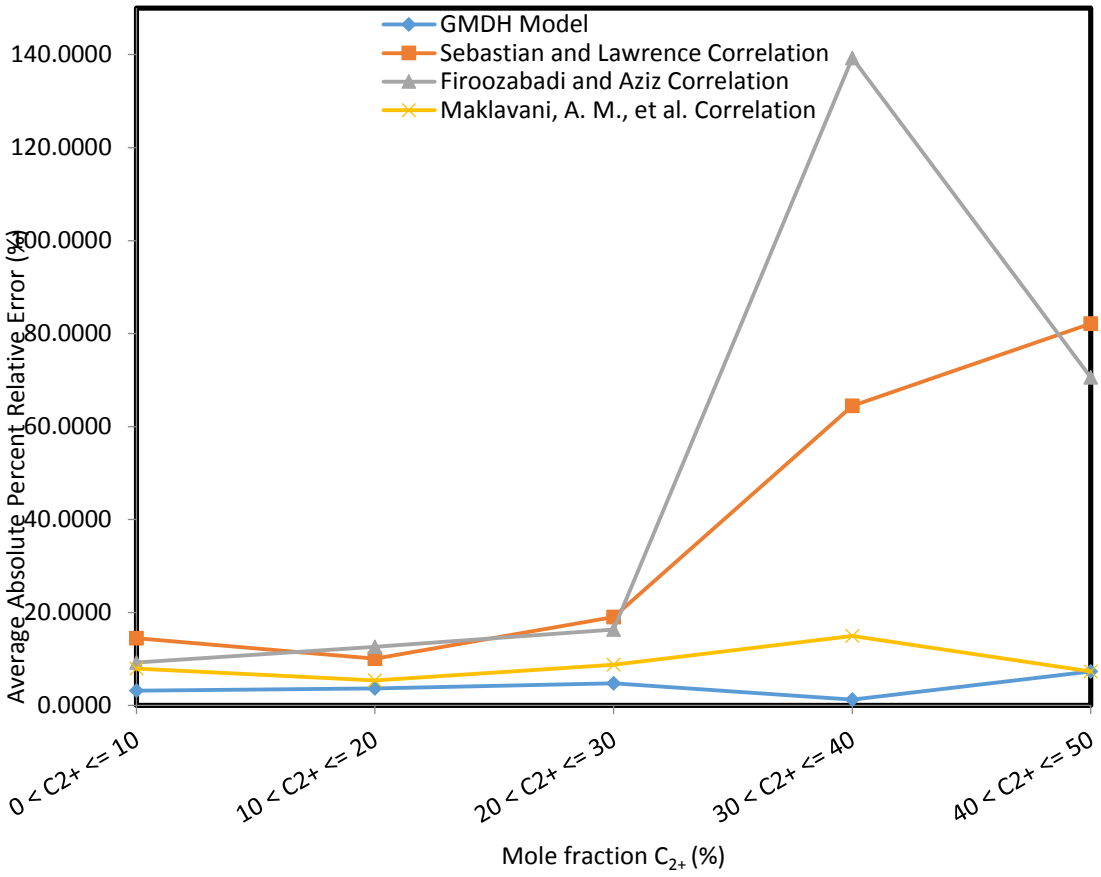


Figure 7: Statistical Accuracy for MMP with Different Mole Fraction of C_{2+} Range

In Figure 7, the models are now compared with the ranges in mole fraction of C_{2+} whereby basically the same trend as before as GMDH recorded lowest value of AAPE in all ranges. In terms of C_{2+} mole fraction ranges, again this signify the performance of GMDH against the previous correlations is better.

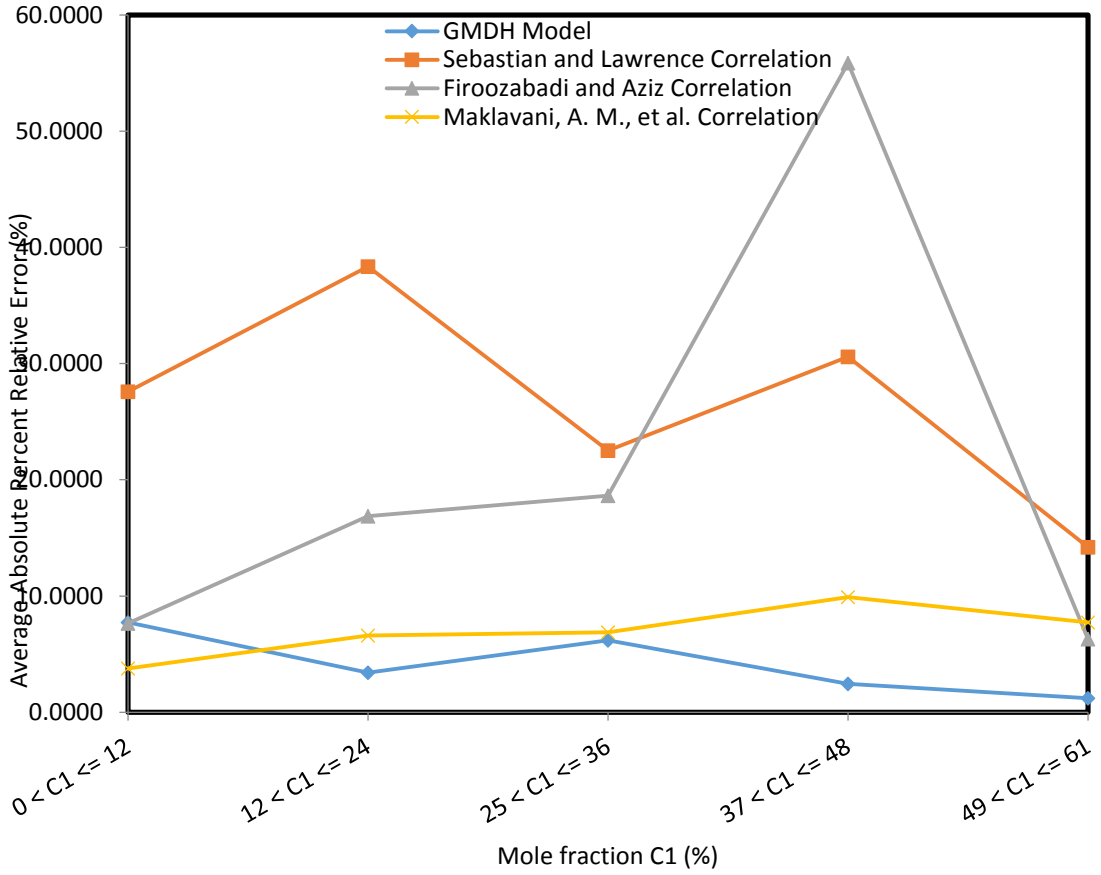


Figure 8: Statistical Accuracy for MMP with Different Mole Fraction of C_1 Range

Referring to Figure 8, comparison of AAPE was made among different ranges of C_1 mole fraction. At range of 0 until 12, Maklavani, A. M., et al. [2] correlation has a better value of AAPE than GMDH. However, for the other ranges, GMDH register a better value of AAPE showing the stability of the model either at lower or higher limit of C1 mole fraction, displaying the lowest value of AAPE.

4.4 Statistical and Graphical Analysis

Table 4 contains results of statistical analysis which include E_r, E_a, E_{min}, E_{max}, RMSE, R and SD to picture the model generated performance.

Table 4: Comparisons of Statistical Analysis for GMDH Model

Parameters	Data Sets		
	Training	Validation	Testing
E_a	4.1395	4.7146	3.6524
E_r	-0.3550	-0.4022	-0.94923
E_{max}	24.3380	21.7938	24.3380
E_{min}	0.0131	0.1874	0.0131
RMSE	6.0077	6.1428	5.1154
R	0.9671	0.9666	0.9822
SD	308.7416	342.7422	268.6085

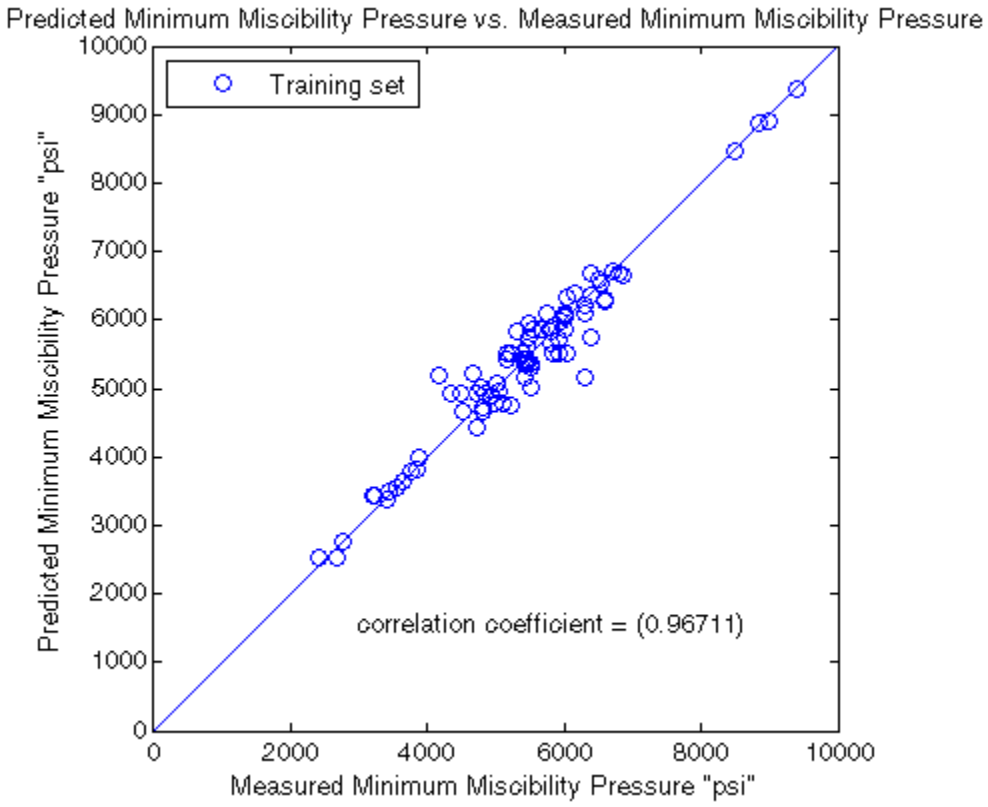


Figure 9: Predicted vs Measured MMP (Training Data Sets)

Figure 9, 10 and 11 shows cross plot for predicted versus measured MMP for training, validation and testing data sets respectively. Figure 9 shows the cross plot for the training data sets of the GMDH model and it produces 0.9671 R value. As per pictured by

the figure, correlation line was set to be ideal at 45° and the tendency of values registered at the middle of the plot suggest that the MMP prediction will be more accurate at the range around 4000 psi to 8000 psi by using GMDH. However, both upper and lower range than that values shows lower registration of MMP.

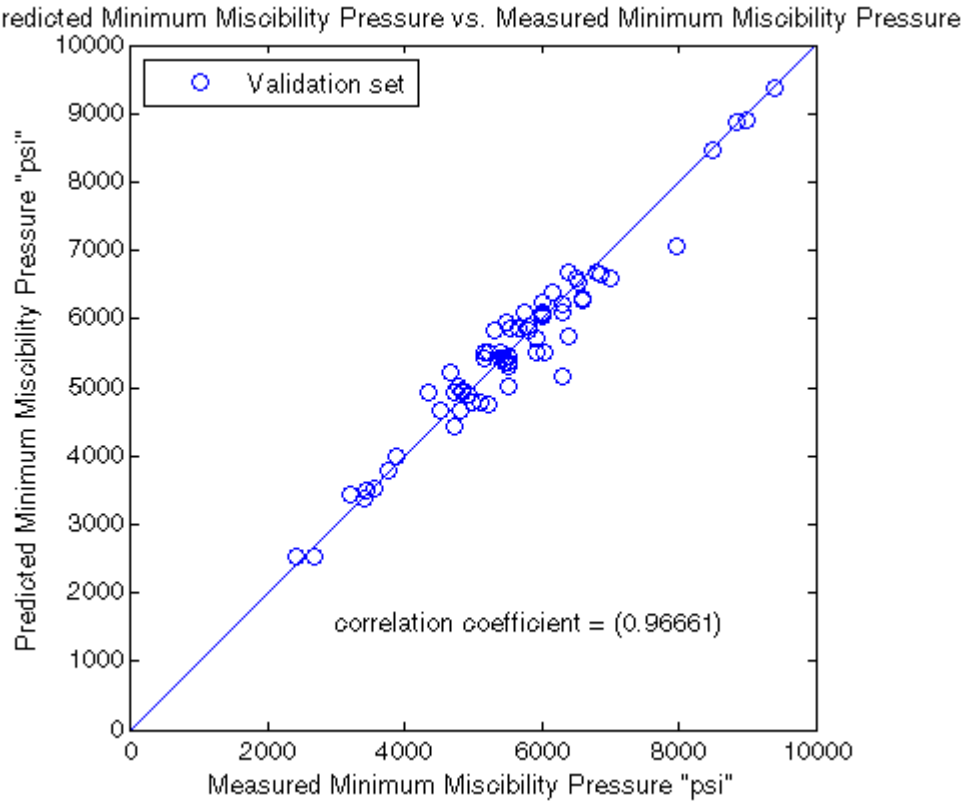


Figure 10: Predicted vs Measured MMP (Validation Data Sets)

In order to avoid overtraining, validation data sets was introduced. Figure 10 shows the cross plot for the validation data sets of the GMDH model and it produces 0.9666 R value, slightly lower than the one in training data sets. Still, the accuracy of MMP determination is still the same as previous; higher in the middle range, lower in the upper and lower range.

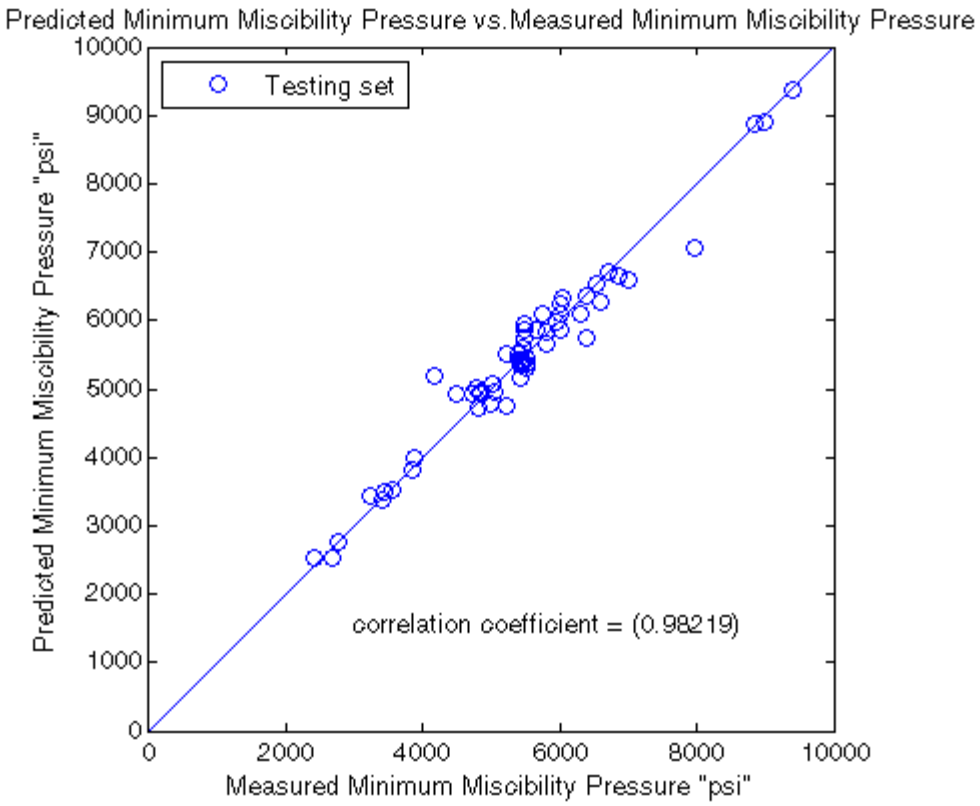


Figure 11: Predicted vs Measured MMP (Testing Data Sets)

Figure 11 shows the cross plot for the testing data sets of the GMDH model and it produces a better correlation coefficient of 0.9822, highest among the data sets. Having the same trend as the earlier data sets, again it shows that MMP prediction is close to ideal at the middle of the graph rather than early and last part of the graph. Even though the value of the correlation coefficient can be a good benchmark in determining the performance of the model, there are other indicators that need to be taken into consideration.

To see the better performance of GMDH model, comparisons were made with the previous correlations. Table 5 summarizes the differences between the models.

Table 5: Comparison of Statistical Analysis

Model Name	Statistical Feature						
	E_a	E_r	E_{max}	E_{min}	RMSE	R	SD
Maklavani, A. M., et al. [2]	4.2830	-2.0624	26.2337	0.0078	6.0873	0.9739	239.2164
Sebastian and Lawrence [4]	24.1412	-9.5583	132.0621	0.4345	37.0458	0.2316	1403.9330
Firoozabadi and Aziz [5]	21.7103	-10.8427	237.6133	0.1781	47.4929	0.4015	1517.7150
GMDH	3.6524	-0.9492	24.3380	0.0131	5.1154	0.9822	268.6085

From Table 5, the author compares the R value of all the models. These coefficients were produced based on the utilization of testing data to each of the correlations. The outcome is GMDH model having the highest R value which is 0.9822 with Maklavani, A. M., et al. [2] model come in next with 0.9739.

On top of that, AAPE value of GMDH model also is the lowest among all. AAPE value can be used as a guidance to have a clearer graphical representation. This made AAPE the main criteria to be look at in this project. With lowest AAPE value of 3.6524%, GMDH model has the superiority as this means that the error from the actual value in this model is the smallest.

The differences between the values of E_{max} and E_{min} shows how far off the prediction values when it is compared with the actual values. If the range is large, the model have a tendency to produce inconsistent predictions. GMDH has the best differences with 24.3380 as the higher limit and 0.0131 as the lower limit followed by Maklavani, A. M., et al. [2] model with the next best error range.

A comparison also done on RMSE value. RMSE value shows the data dispersion around the zero deviation. Based on table 5, GMDH model has the lowest value of RMSE at 5.1154% followed by Maklavani, A. M., et al. [2] model at 6.0873%. This signify that GMDH model has less dispersion around the zero deviation compared to other correlations.

Comparing the SD of the models, Maklavani, A. M., et al. [2] model has the lowest SD value which is 239.2164 psi compared to GMDH model, 268.6085, in the second place. SD also signify the dispersion of values obtained by using that particular model. Thus, this time Maklavani, A. M., et al. [2] model has lower dispersion compared to GMDH model.

To rank each of the performances, Table 6 and 7 shows the comparison in terms of AAPE and R value, and RMSE and SD value respectively.

Table 6: Model Ranking based on E_a and R value

Model Name	Statistical Feature		Rank
	E_a	R	
GMDH Model	3.6524	0.9822	1
Maklavani, A. M., et al. model [2]	4.2830	0.9739	2
Firoozabadi and Aziz [5]	21.7103	0.4015	3
Sebastian and Lawrence [4]	24.1412	0.2316	4

Table 7: Model Ranking based on RMSE and SD value

Model Name	Statistical Feature		Rank
	RMSE	SD	
GMDH Model	5.1154	268.6085	1
Maklavani, A. M., et al. model [2]	6.0873	239.2164	2
Sebastian and Lawrence [4]	37.0458	1403.9330	3
Firoozabadi and Aziz [5]	47.4929	1517.7150	4

Ranking was decided in Table 6 based on AAPE and R value due to the nature of the statistical analysis. While the closer R value get to 1, AAPE will confirm its reliability by registering a low value of error. With GMDH model having the lowest E_a value and the highest R value, GMDH is ranked at first place followed by Maklavani, A. M., et al. [2] model at second place.

However, it is a close call in Table 7, differentiating the RMSE and SD values of the models. Despite having a higher SD value, GMDH model also ranked at first place ahead of Maklavani, A. M., et al. [2] model. This is due to the lower RMSE value which contribute in having a lower dispersion beside the superiority on the early comparison above. Therefore, both GMDH model and Maklavani, A. M., et al. [2] model have lower dispersion but GMDH only ahead by tiny differences.

As an overview, it can be seen from the tables above that GMDH model gives good MMP prediction as it has low AAPE, RMSE and SD value together with the highest R value. These shows that the model produce less errors and results that are far out from the reasonable range.

Figure 12, 13 and 14 shows the error distribution for training, validation and testing data sets respectively.

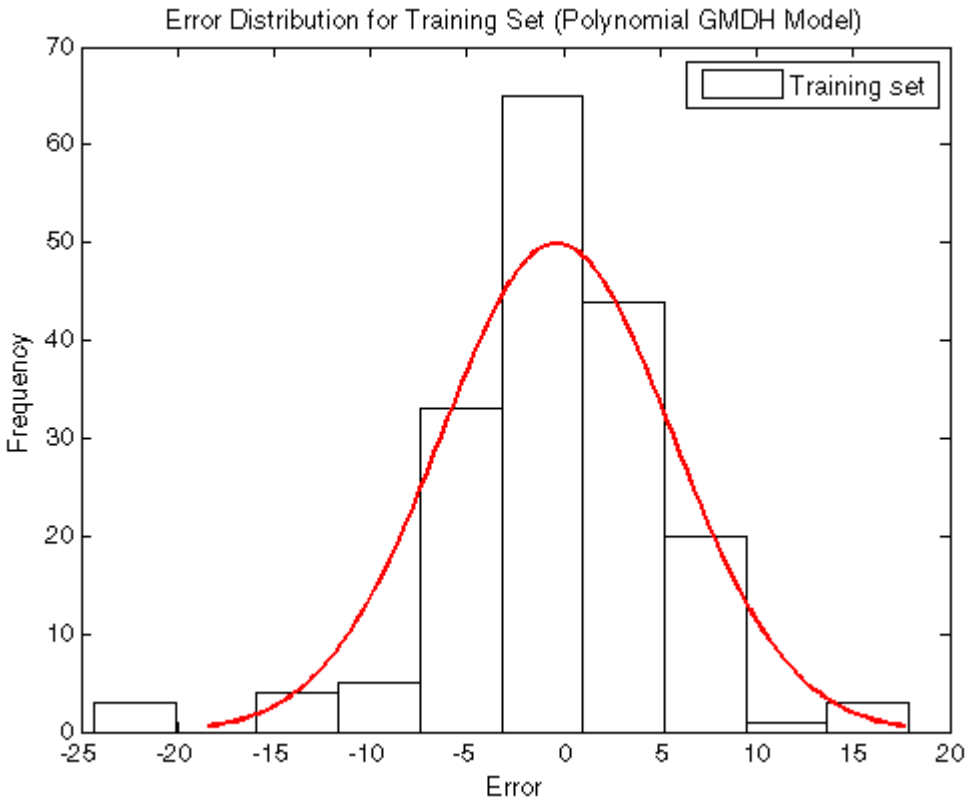


Figure 12: Error Distribution Graph (Training Data Sets)

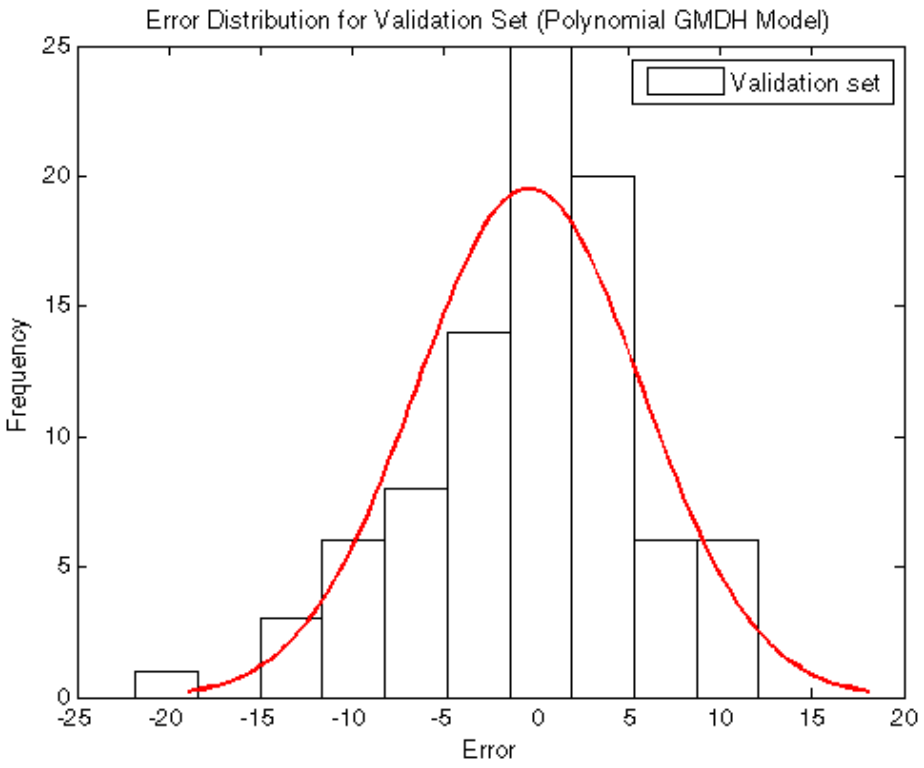


Figure 13: Error Distribution Graph (Validation Data Sets)

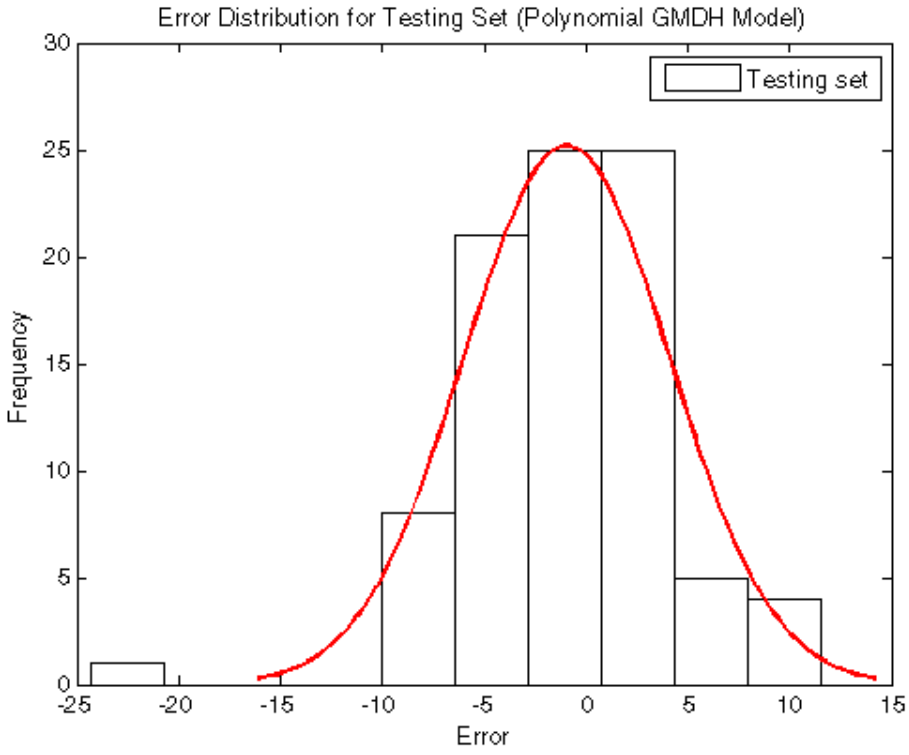


Figure 14: Error Distribution Graph (Testing Data Sets)

From Figure 12, there is a slight shift to the left of the plotted means of errors, negatively valued. In Figure 13, the peaks also shift to the left of the plotted means of errors, again bearing negative error while in Figure 14, the means of errors distributed to the right of the plot. This 3 figures signify that there is no fixed trends of error distribution as all plots peaks nearly towards the zero error.

To check for model consistency, residual analysis was utilized. In this analysis, the difference between the predicted and measured values is plotted with a zero line to indicate where the error values located at. Figure 15, 16 and 17 shows the residual graph for training, validation and testing data sets respectively.

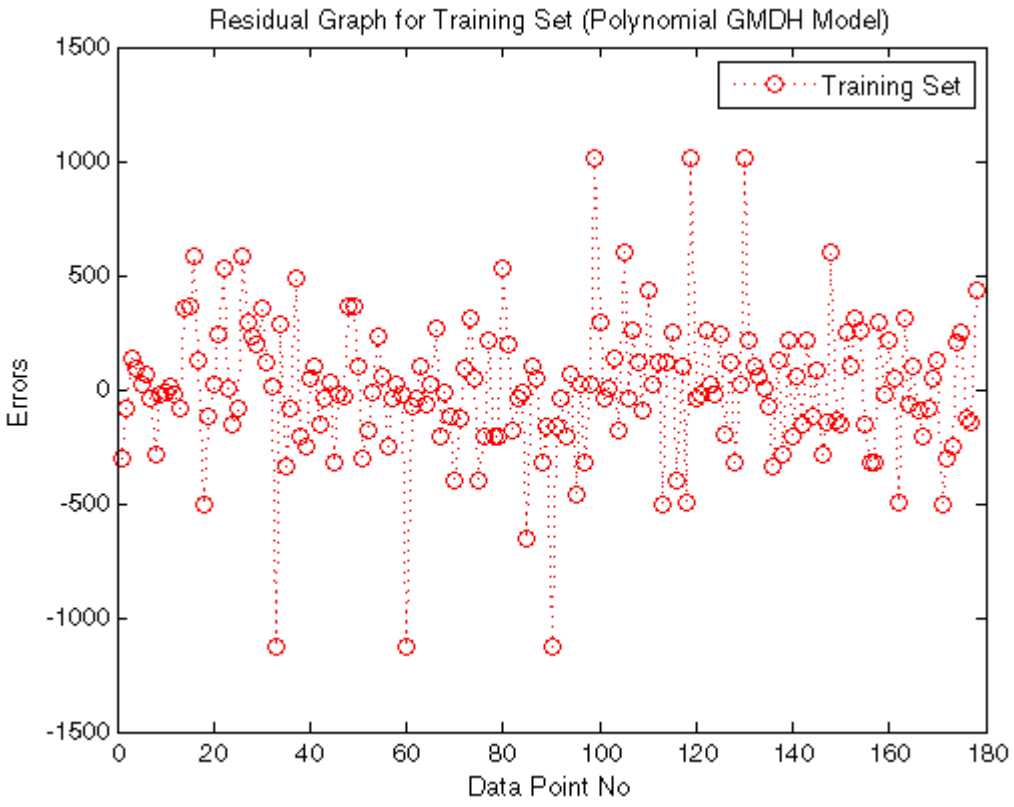


Figure 15: Residual Graph (Testing Data Sets)

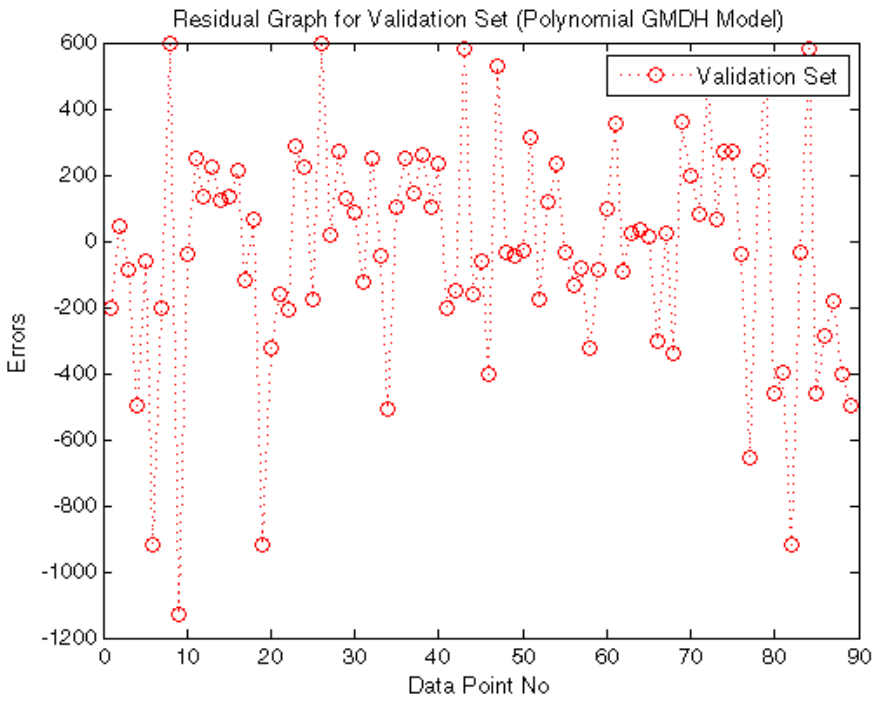


Figure 16: Residual Graph (Validation Data Sets)

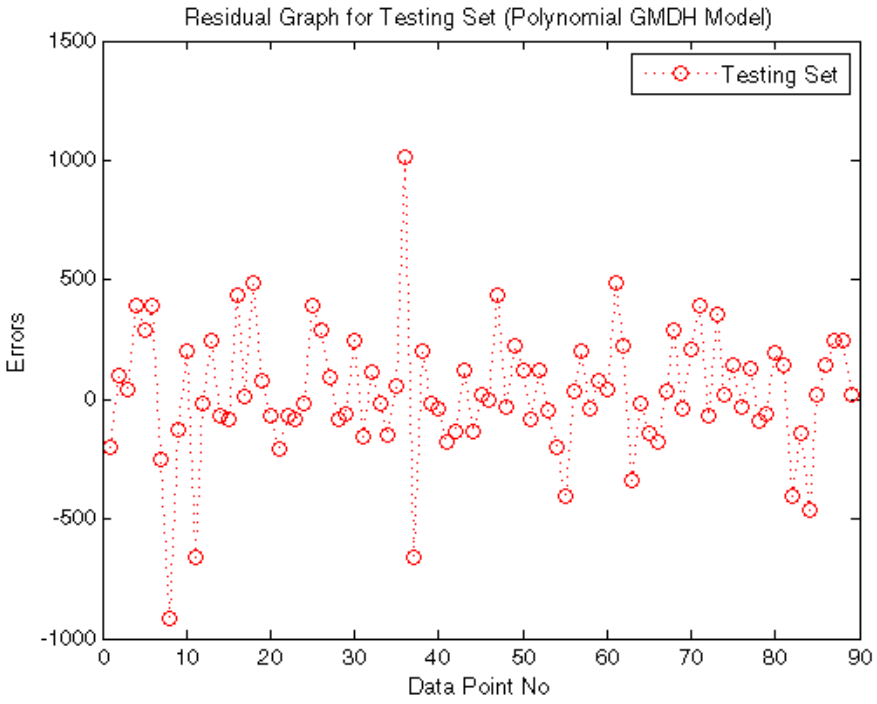


Figure 17: Residual Graph (Validation Data Sets)

From Figure 15, the minimum and maximum residual errors for training data are -1127.32 and 1015.868 respectively. From Figure 16, the minimum and maximum residual errors for validation data are -1127.32 and 596.8538 respectively. From Figure 17, the minimum and maximum residual errors for testing data are -919.452 and 1015.868 respectively. This 3 figures also signify that there is no obvious tendency towards the negative and positive sides of errors.

To verify the idealistic trend of the parameter involved in the study, trend analysis was conducted. Table 8 show the data range used in determination of trend analysis.

Table 8: Data Ranges for Trend Analysis

Data Range	Temperature	Molecular weight of C_{7+}	Mole fraction of intermediates ($C_2 – C_6$)	Mole fraction of C_1	Mole fraction of C_{2+}
Min	121.9100	121.9100	1.0000	0.0000	0.0000
Max	340.0000	302.0000	36.9000	60.5500	47.0000

Figure 18, 19 and 20 shows the trend of selected parameters towards the MMP.

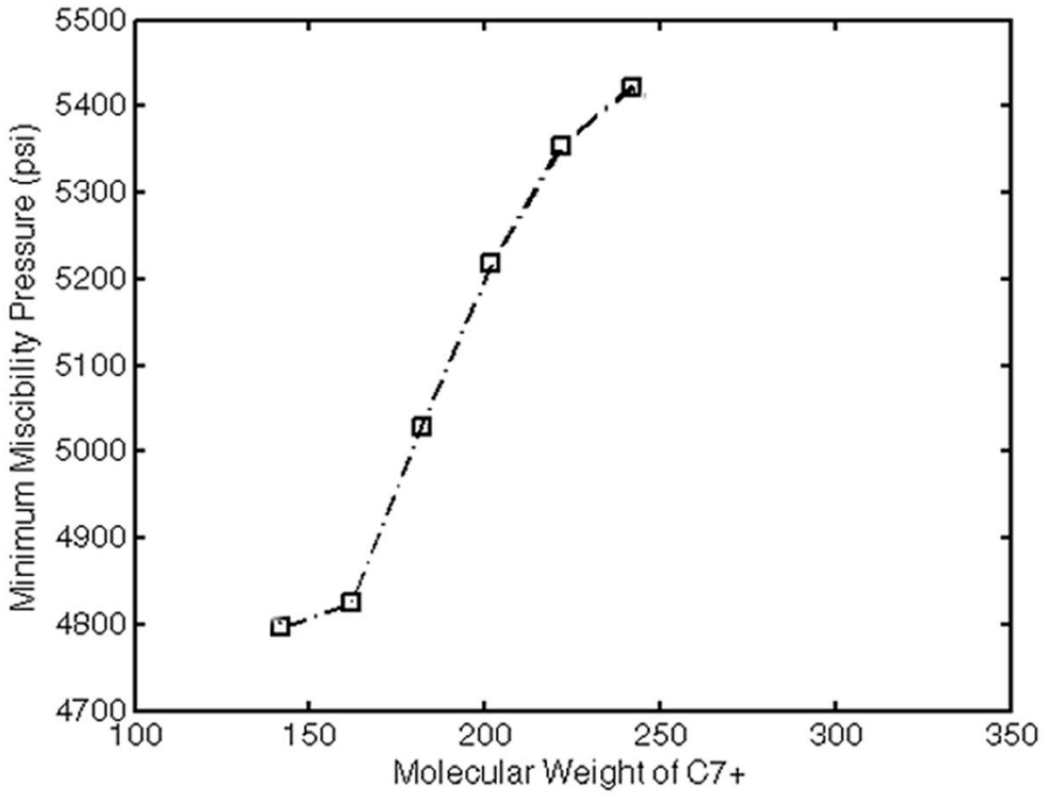


Figure 18: Molecular Weight of C_{7+} vs MMP

Figure 18 shows the trend of molecular weight of C_{7+} against the predicted MMP. It shows that as the molecular weight of C_{7+} increase, the MMP also increase. Based on study by Maklavani, A. M., et al. [2], they also stated the same trend. This conclude that the molecular weight of C_{7+} trend adhere to the real behavior.

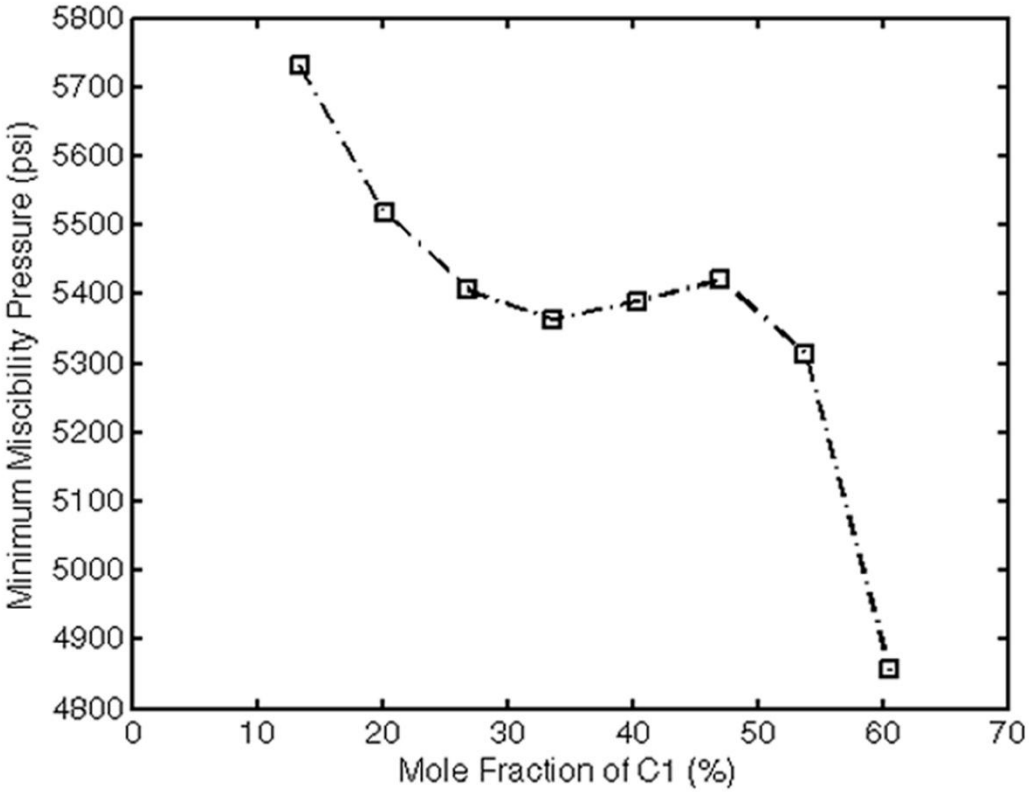


Figure 19: Mole Fraction of C_1 vs MMP

Figure 19 shows the trend of mole fraction of C_1 against the predicted MMP. It shows that as the mole fraction of C_1 increase, the MMP decrease. Based on study by Maklavani, A. M., et al. [2], they also stated the same trend. This conclude that the mole fraction of C_1 trend adhere to the real behavior.

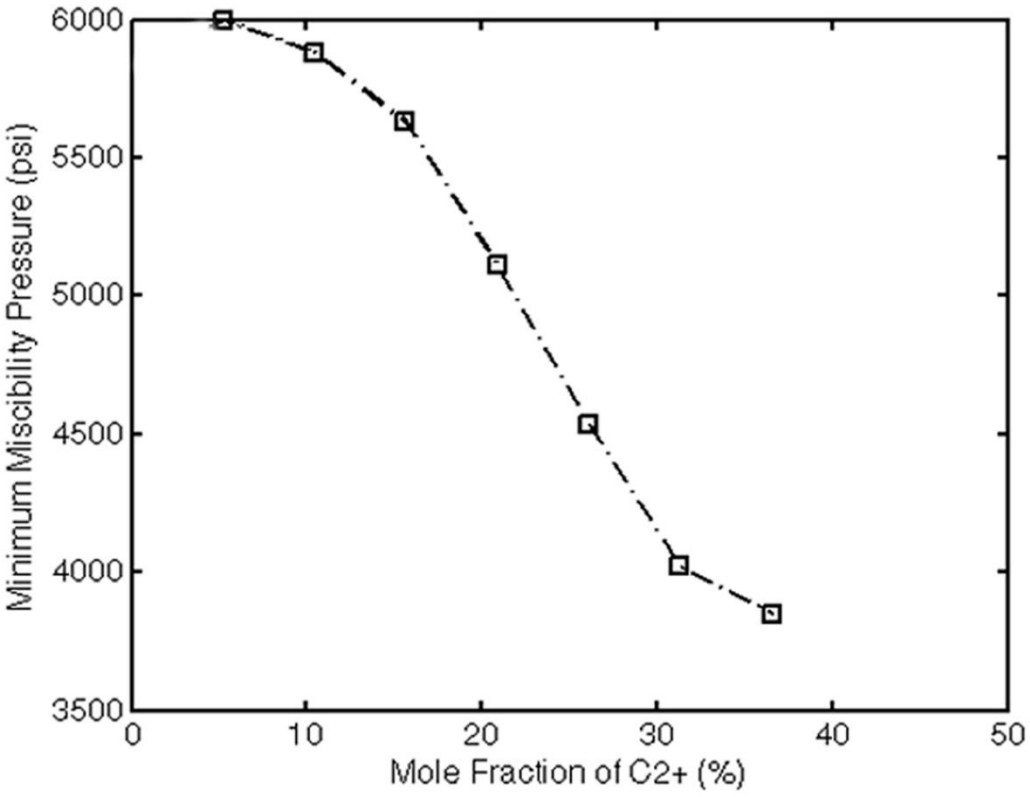


Figure 20: Mole Fraction of C_{2+} vs MMP

Figure 20 shows the trend of mole fraction of C_{2+} against the predicted MMP. It shows that as the mole fraction of C_{2+} increase, the MMP decrease. Based on study by Maklavani, A. M., et al. [2], they also stated the same trend. This conclude that the mole fraction of C_{2+} trend adhere to the real behavior.

CHAPTER 5
CONCLUSION AND RECOMMENDATION

As a conclusion, this study is focusing on the prediction of MMP by utilizing the GMDH method. Proposed method to determine MMP using GMDH method was successfully conducted using common input parameters which are reservoir temperature, molecular weight of C_{7+} in the oil, mole fraction of intermediates ($C_2 – C_6$) in the oil, mole fraction of C_1 in the oil and mole fraction of C_{2+} in the injected gas. However, only mole fraction of intermediates ($C_2 – C_6$) in the oil, mole fraction of C_1 in the oil and mole fraction of C_{2+} in the injected gas found to be most influential to MMP prediction. Statistical and graphical analysis done enable the author to test and validate the performance of the model as low E_a, RMSE and SD together with high R value were obtained. Statistical comparison made with the previous correlations also showing that GMDH has better performance as E_a recorded was 3.6524 %, lowest among all and signify less errors can be found. The objectives of the study were achieved.

There are several improvements can be made towards the betterment of this study. The author believe an increment in data sets will increase the accuracy of the results thus better correlation can be produced. Other sources of data should be discovered as the author believe that simulation project with enormous of data will generate better correlation. The author would also like to suggest this model to be implemented in the industry. This will give a larger overview on how the performance of the model and improvisation still can be made to the current model produced.

REFERENCES

[1] G. Chen, X. Wang, Z. Liang, R. Gao, T. Sema, P. Luo, *et al.*, "Simulation of CO2-Oil Minimum Miscibility Pressure (MMP) for CO2 Enhanced Oil Recovery (EOR) using Neural Networks," *Energy Procedia,* vol. 37, pp. 6877-6884, // 2013.

[2] A. M. Maklavani, A. Vatani, B. Moradi, and J. Tangsirifard, "New Minimum Miscibiity Pressure (MMP) Correlation for Hydrocarbon Miscible Injections," *Brazilian Journal of Petroleum and Gas,* vol. 4, pp. 11-19, 2010.

[3] O. Glass, "Generalized Minimum Miscibility Pressure Correlation (includes associated papers 15845 and 16287)," 1985/12/1/.

[4] H. M. Sebastian and D. D. Lawrence, "Nitrogen Minimum Miscibility Pressures."

[5] A. Firoozabadi and A. Khalid, "Analysis and Correlation of Nitrogen and Lean-Gas Miscibility Pressure (includes associated paper 16463)," 1986/11/1/.

[6] K. Fujimoto and S. Nakabayashi, "Applying GMDH algorithm to extract rules from examples," *Syst. Anal. Model. Simul.,* vol. 43, pp. 1311-1319, 2003.

[7] V. P. Astakhov and V. V. Galitsky, "Tool life testing in gundrilling: an application of the group method of data handling (GMDH)," *International Journal of Machine Tools and Manufacture,* vol. 45.

[8] A. A. Semenov, R. A. Oshmarin, A. Driller, and A. Butakova, "Application of Group Method of Data Handling for Geological Modeling of Vankor Field."

[9] B. Y. Lee, H. S. Liu, and Y. S. Tarng, "An Abductive Network for Predicting Tool Life in Drilling," vol. 35.

[10] F. M. Orr, Jr. and M. K. Silva, "Effect of Oil Composition on Minimum Miscibility Pressure-Part 2: Correlation," 1987/11/1/.

[11] P. Kordik, P. Naplava, M. Snorek, and M. Genyk-Berezovskyj, "Modified GMDH Method and Models Quality Evaluation by Visualization."

[12] T. H. Ahmed, "Prediction of CO2 Minimum Miscibility Pressures."

[13] S. M. Seyyedsar, M. Assareh, and C. Ghotbi, "Minimum Miscibility Pressure Calculation: A Comparison between Multi Contact Test and Compositional

Simulation."

[14] Y. Wang and F. M. Orr, Jr., "Calculation of Minimum Miscibility Pressure."

[15] M. R. Akbari and N. Kasiri, "Determination of Minimum Miscibility Pressure in Gas Injection Process by Using ANN with Various Mixing Rules."

APPENDIX A

Data set

MMP (psi)	Temperature (°F)	MW C7+	Mole fraction C2 - C6 (%)	Mole fraction C1 (%)	Mole fraction C2+ (%)
4718.9478	210.0020	252.2100	25.0000	32.0000	30.0000
5050.0000	280.0000	215.0000	34.9800	24.6800	0.0000
4531.9942	252.2120	252.2100	25.0000	32.0000	28.0000
4850.0000	225.0000	140.0000	7.6300	38.6200	0.0000
5400.4802	240.6740	240.6800	25.0000	46.0000	18.0000
6000.0662	249.9980	250.0000	23.0000	50.0000	10.0000
5456.0296	240.6740	240.6800	25.0000	46.0000	18.0000
6598.0568	269.9960	270.0000	27.0000	31.0000	0.0000
3565.0276	174.5960	175.0000	28.0000	6.0000	43.0000
3650.0197	302.0000	302.0000	1.0000	42.0000	38.0000
6526.0000	268.0000	223.0000	16.0700	57.9500	0.0000
3565.0276	174.5960	175.0000	28.0000	6.0000	43.0000
9000.0000	300.0000	140.0000	3.7200	9.3700	0.0000
5753.9371	252.2120	252.2100	25.0000	32.0000	5.0000
5161.0229	209.8040	209.8100	26.0000	33.0000	16.0000
4650.0000	196.0000	232.0000	32.2000	43.0000	0.0000
2399.9394	215.0060	215.0000	24.0000	33.0000	47.0000
6028.9287	209.8040	209.8100	26.0000	33.0000	16.0000
6800.0000	280.0000	261.0000	27.0500	21.5400	0.0000
2750.0605	302.0000	302.0000	1.0000	42.0000	46.0000
5489.9684	210.0020	252.2100	25.0000	32.0000	15.0000
5309.9766	245.4260	245.4300	30.2500	26.5700	12.0000
6700.0000	279.0000	140.0000	5.5000	22.6900	0.0000
2680.0073	215.0060	215.0000	24.0000	33.0000	47.0000
6300.0000	303.0000	250.0000	22.8000	49.0100	0.0000
4650.0000	196.0000	232.0000	32.2000	43.0000	0.0000
6029.9439	252.2120	252.2100	25.0000	32.0000	0.0000
3206.0592	257.7020	257.7000	32.0800	22.9200	41.4700
4728.0000	260.0000	190.0000	27.7900	57.8400	0.0000
5753.9371	252.2120	252.2100	25.0000	32.0000	5.0000
5467.0000	260.0000	199.0000	25.0600	60.3600	0.0000
6526.0000	268.0000	223.0000	16.0700	57.9500	0.0000
6300.0042	183.5960	183.6000	26.0000	57.0000	0.0000
5228.7555	121.9100	121.9100	26.0000	33.0000	15.0000
6600.0000	199.0000	232.0000	19.6000	36.2000	0.0000
6300.0000	303.0000	250.0000	22.8000	49.0100	0.0000

5475.1746	210.0020	252.2100	25.0000	32.0000	10.0000
6300.0042	249.9980	250.0000	23.0000	49.0000	0.0000
5427.0221	269.9960	270.0000	27.0000	31.0000	20.0000
5946.0000	268.0000	218.0000	20.6200	54.6300	0.0000
5800.0591	208.9940	209.0000	22.0000	55.0000	14.0000
6000.0000	280.0000	261.0000	33.6300	36.7800	0.0000
8500.0000	225.0000	140.0000	3.7200	9.3700	0.0000
5800.0000	200.0000	209.0000	22.1600	54.5000	15.3700
5845.0208	209.8040	209.8100	26.0000	33.0000	16.0000
3840.0000	250.0000	195.0000	36.9000	32.7300	0.0000
9400.0000	279.0000	140.0000	2.4700	0.0000	0.0000
5161.0229	209.8040	209.8100	26.0000	33.0000	16.0000
5161.0229	209.8040	209.8100	26.0000	33.0000	16.0000
6000.0662	249.9980	250.0000	22.0000	50.0000	10.0000
4718.9478	210.0020	252.2100	25.0000	32.0000	30.0000
5907.9672	269.9960	270.0000	27.0000	31.0000	10.0000
3650.0197	302.0000	302.0000	1.0000	42.0000	38.0000
3206.0592	257.7020	257.7000	32.0800	22.9200	41.4700
5000.0310	294.9620	294.9700	25.0000	37.0000	23.0000
5427.0221	269.9960	270.0000	27.0000	31.0000	20.0000
3408.9670	254.4080	254.4000	30.3600	23.6400	41.4700
8850.0000	279.0000	140.0000	3.7200	9.3700	0.0000
3565.0276	174.5960	175.0000	28.0000	6.0000	43.0000
6300.0042	183.5960	183.6000	26.0000	57.0000	0.0000
5423.9763	210.0020	252.2100	25.0000	32.0000	20.0000
5456.0296	240.6740	240.6800	25.0000	46.0000	18.0000
6500.0000	250.0000	261.0000	27.0500	21.5400	0.0000
4800.0000	340.0000	183.6000	26.3700	56.8600	0.0000
3754.0000	140.0000	191.0000	32.7000	42.7000	0.0000
6400.0000	225.0000	140.0000	5.5000	22.6900	0.0000
4980.0000	279.0000	140.0000	7.6300	38.6200	0.0000
3650.0197	302.0000	302.0000	1.0000	42.0000	38.0000
6800.0000	280.0000	261.0000	27.0500	21.5400	0.0000
5923.0511	209.8040	209.8100	26.0000	33.0000	16.0000
5500.0000	280.0000	230.0000	34.0500	34.6400	0.0000
4850.0000	225.0000	140.0000	7.6300	38.6200	0.0000
5551.0293	252.2120	252.2100	25.0000	32.0000	10.0000
5946.0000	268.0000	218.0000	20.6200	54.6300	0.0000
5923.0511	209.8040	209.8100	26.0000	33.0000	16.0000
6300.0042	249.9980	250.0000	23.0000	49.0000	0.0000
6164.0000	268.0000	221.0000	18.2300	57.3600	0.0000

4980.0000	279.0000	140.0000	7.6300	38.6200	0.0000
6300.0042	249.9980	250.0000	23.0000	49.0000	0.0000
5309.9766	245.4260	245.4300	30.2500	26.5700	12.0000
4728.0000	260.0000	190.0000	27.7900	57.8400	0.0000
5907.9672	269.9960	270.0000	27.0000	31.0000	10.0000
4924.0312	210.0020	252.2100	25.0000	32.0000	25.0000
3650.0197	302.0000	302.0000	1.0000	42.0000	38.0000
6400.0000	140.0000	232.0000	19.6000	36.2000	0.0000
5800.0591	208.9940	209.0000	22.0000	55.0000	14.0000
6000.0000	225.0000	250.0000	22.4900	50.3900	12.1500
5100.0000	300.0000	140.0000	7.6300	38.6200	0.0000
4814.9628	179.9960	197.3000	22.0000	50.0000	25.0000
6300.0042	183.5960	183.6000	26.0000	57.0000	0.0000
4814.9628	179.9960	197.3000	22.0000	50.0000	25.0000
4924.0312	210.0020	252.2100	25.0000	32.0000	25.0000
6300.0042	249.9980	250.0000	23.0000	49.0000	0.0000
6000.0662	249.9980	250.0000	23.0000	50.0000	10.0000
5214.6868	231.0080	231.0000	31.0000	44.0000	27.0000
8850.0000	279.0000	140.0000	3.7200	9.3700	0.0000
5845.0208	209.8040	209.8100	26.0000	33.0000	16.0000
5400.4802	240.6740	240.6800	25.0000	46.0000	18.0000
4174.0000	164.0000	193.0000	25.1700	52.6200	0.0000
6029.9439	252.2120	252.2100	25.0000	32.0000	0.0000
8500.0000	225.0000	140.0000	3.7200	9.3700	0.0000
6700.0000	279.0000	140.0000	5.5000	22.6900	0.0000
4531.9942	252.2120	252.2100	25.0000	32.0000	28.0000
5497.9455	238.1540	238.1500	24.1200	54.2600	17.8800
4336.0000	279.0000	179.0000	23.0900	45.2300	0.0000
4924.0312	210.0020	252.2100	25.0000	32.0000	25.0000
5157.9771	240.6740	240.6800	25.0000	46.0000	18.0000
5467.0000	260.0000	199.0000	25.0600	60.3600	0.0000
5455.0144	143.7080	143.7000	24.6800	45.8500	18.1200
4500.0000	176.0000	192.0000	14.5500	13.0000	0.0000
2750.0605	302.0000	302.0000	1.0000	42.0000	46.0000
5407.0069	209.8040	209.8100	26.0000	33.0000	16.0000
6028.9287	209.8040	209.8100	26.0000	33.0000	16.0000
5467.0000	260.0000	199.0000	25.0600	60.3600	0.0000
4774.9324	271.0040	271.0000	27.0000	31.0000	23.0000
5923.0511	209.8040	209.8100	26.0000	33.0000	16.0000
6000.0662	249.9980	250.0000	22.0000	50.0000	10.0000
5500.0000	225.0000	215.0000	34.9800	24.6800	0.0000

4174.0000	164.0000	193.0000	25.1700	52.6200	0.0000
8500.0000	225.0000	140.0000	3.7200	9.3700	0.0000
3840.0000	250.0000	195.0000	36.9000	32.7300	0.0000
5157.9771	240.6740	240.6800	25.0000	46.0000	18.0000
3754.0000	140.0000	191.0000	32.7000	42.7000	0.0000
6381.0000	268.0000	221.0000	16.5000	56.6500	0.0000
5489.9684	210.0020	252.2100	25.0000	32.0000	15.0000
6850.0000	300.0000	140.0000	5.5000	22.6900	0.0000
5407.0069	209.8040	209.8100	26.0000	33.0000	16.0000
5845.0208	209.8040	209.8100	26.0000	33.0000	16.0000
3754.0000	140.0000	191.0000	32.7000	42.7000	0.0000
4174.0000	164.0000	193.0000	25.1700	52.6200	0.0000
5652.9909	252.2120	252.2100	25.0000	32.0000	10.0000
6500.0000	250.0000	261.0000	27.0500	21.5400	0.0000
5000.0310	294.9620	294.9700	25.0000	37.0000	23.0000
6700.0000	279.0000	140.0000	5.5000	22.6900	0.0000
5423.9763	210.0020	252.2100	25.0000	32.0000	20.0000
6600.0000	199.0000	232.0000	19.6000	36.2000	0.0000
2399.9394	215.0060	215.0000	24.0000	33.0000	47.0000
6598.0568	269.9960	270.0000	27.0000	31.0000	0.0000
5652.9909	252.2120	252.2100	25.0000	32.0000	10.0000
6300.0042	249.9980	250.0000	23.0000	49.0000	0.0000
5000.0310	294.9620	294.9700	25.0000	37.0000	23.0000
6000.0000	280.0000	261.0000	33.6300	36.7800	0.0000
6164.0000	268.0000	221.0000	18.2300	57.3600	0.0000
6800.0000	280.0000	261.0000	27.0500	21.5400	0.0000
3423.0356	141.9980	141.9900	30.0000	24.0000	41.0000
6598.0568	269.9960	270.0000	27.0000	31.0000	0.0000
5800.0591	208.9940	209.0000	22.0000	55.0000	16.0000
4336.0000	279.0000	179.0000	23.0900	45.2300	0.0000
5500.0000	140.0000	232.0000	20.9000	54.7000	0.0000
2680.0073	215.0060	215.0000	24.0000	33.0000	47.0000
4774.9324	271.0040	271.0000	27.0000	31.0000	23.0000
5800.0591	208.9940	209.0000	22.0000	55.0000	14.0000
5551.0293	252.2120	252.2100	25.0000	32.0000	10.0000
5157.9771	240.6740	240.6800	25.0000	46.0000	18.0000
6000.0000	280.0000	261.0000	33.6300	36.7800	0.0000
5845.0208	209.8040	209.8100	26.0000	33.0000	16.0000
5100.0000	300.0000	140.0000	7.6300	38.6200	0.0000
6029.9439	252.2120	252.2100	25.0000	32.0000	0.0000
3565.0276	174.5960	175.0000	28.0000	6.0000	43.0000

6164.0000	268.0000	221.0000	18.2300	57.3600	0.0000
6000.0000	225.0000	250.0000	22.4900	50.3900	12.1500
5500.0000	225.0000	215.0000	34.9800	24.6800	0.0000
5551.0293	252.2120	252.2100	25.0000	32.0000	10.0000
4800.0000	340.0000	183.6000	26.3700	56.8600	0.0000
6500.0000	250.0000	261.0000	27.0500	21.5400	0.0000
5455.0144	143.7080	143.7000	24.6800	45.8500	18.1200
6300.0042	249.9980	250.0000	23.0000	49.0000	0.0000
5050.0000	280.0000	215.0000	34.9800	24.6800	0.0000
6000.0000	225.0000	250.0000	22.4900	50.3900	12.1500
3880.0496	302.0000	302.0000	1.0000	42.0000	35.0000
6028.9287	209.8040	209.8100	26.0000	33.0000	16.0000
4718.9478	210.0020	252.2100	25.0000	32.0000	30.0000
5427.0221	269.9960	270.0000	27.0000	31.0000	20.0000
3219.9828	141.7460	141.7400	33.0000	23.0000	41.0000
4774.9324	271.0040	271.0000	27.0000	31.0000	23.0000
5500.0000	280.0000	230.0000	34.0500	34.6400	0.0000
5800.0591	208.9940	209.0000	22.0000	55.0000	16.0000
4500.0000	176.0000	192.0000	14.5500	13.0000	0.0000
6850.0000	300.0000	140.0000	5.5000	22.6900	0.0000
6000.0000	225.0000	250.0000	22.4900	50.3900	12.1500
6300.0000	303.0000	250.0000	22.8000	49.0100	0.0000
5500.0000	225.0000	215.0000	34.9800	24.6800	0.0000
5511.5791	132.1160	132.1200	24.0000	54.0000	17.0000
7977.0000	268.0000	244.0000	14.0000	60.5500	0.0000
6300.0042	249.9980	250.0000	23.0000	49.0000	0.0000
4336.0000	279.0000	179.0000	23.0900	45.2300	0.0000
6300.0042	183.5960	183.6000	26.0000	57.0000	0.0000
3549.9437	249.9980	274.5000	29.7500	26.1500	43.0700
4774.9324	271.0040	271.0000	27.0000	31.0000	23.0000
4531.9942	252.2120	252.2100	25.0000	32.0000	28.0000
6017.0000	271.0000	232.0000	20.9000	54.7000	0.0000
3880.0496	302.0000	302.0000	1.0000	42.0000	35.0000
4531.9942	252.2120	252.2100	25.0000	32.0000	28.0000
6164.0000	268.0000	221.0000	18.2300	57.3600	0.0000
6800.0000	280.0000	261.0000	27.0500	21.5400	0.0000
6000.0662	249.9980	250.0000	23.0000	50.0000	10.0000
7977.0000	268.0000	244.0000	14.0000	60.5500	0.0000
5100.0000	300.0000	140.0000	7.6300	38.6200	0.0000
4814.9628	179.9960	197.3000	22.0000	50.0000	25.0000
4980.0000	279.0000	140.0000	7.6300	38.6200	0.0000

5228.7555	121.9100	121.9100	26.0000	33.0000	15.0000
6017.0000	271.0000	232.0000	20.9000	54.7000	0.0000
5907.9672	269.9960	270.0000	27.0000	31.0000	10.0000
4336.0000	279.0000	179.0000	23.0900	45.2300	0.0000
5400.4802	240.6740	240.6800	25.0000	46.0000	18.0000
6400.0000	225.0000	140.0000	5.5000	22.6900	0.0000
2399.9394	215.0060	215.0000	24.0000	33.0000	47.0000
4850.0000	225.0000	140.0000	7.6300	38.6200	0.0000
5500.0000	280.0000	230.0000	34.0500	34.6400	0.0000
4774.9324	271.0040	271.0000	27.0000	31.0000	23.0000
3408.9670	254.4080	254.4000	30.3600	23.6400	41.4700
6028.9287	209.8040	209.8100	26.0000	33.0000	16.0000
5800.0591	208.9940	209.0000	22.0000	55.0000	14.0000
4774.9324	271.0040	271.0000	27.0000	31.0000	23.0000
4825.9857	258.0080	258.0000	27.0000	39.0000	24.0000
5157.9771	240.6740	240.6800	25.0000	46.0000	18.0000
6500.0000	250.0000	261.0000	27.0500	21.5400	0.0000
3206.0592	257.7020	257.7000	32.0800	22.9200	41.4700
6850.0000	300.0000	140.0000	5.5000	22.6900	0.0000
2680.0073	215.0060	215.0000	24.0000	33.0000	47.0000
4650.0000	196.0000	232.0000	32.2000	43.0000	0.0000
4814.9628	179.9960	197.3000	22.0000	50.0000	25.0000
5511.5791	132.1160	132.1200	24.0000	54.0000	17.0000
7000.0000	300.0000	261.0000	27.0500	21.5400	0.0000
5309.9766	245.4260	245.4300	30.2500	26.5700	12.0000
4924.0312	210.0020	252.2100	25.0000	32.0000	25.0000
3408.9670	254.4080	254.4000	30.3600	23.6400	41.4700
9400.0000	279.0000	140.0000	2.4700	0.0000	0.0000
5551.0293	252.2120	252.2100	25.0000	32.0000	10.0000
5907.9672	269.9960	270.0000	27.0000	31.0000	10.0000
5407.0069	209.8040	209.8100	26.0000	33.0000	16.0000
3206.0592	257.7020	257.7000	32.0800	22.9200	41.4700
8500.0000	225.0000	140.0000	3.7200	9.3700	0.0000
5500.0000	140.0000	232.0000	20.9000	54.7000	0.0000
9000.0000	300.0000	140.0000	3.7200	9.3700	0.0000
5100.0000	300.0000	140.0000	7.6300	38.6200	0.0000
6300.0000	303.0000	250.0000	22.8000	49.0100	0.0000
6000.0662	249.9980	250.0000	22.0000	50.0000	10.0000
5753.9371	252.2120	252.2100	25.0000	32.0000	5.0000
5455.0144	143.7080	143.7000	24.6800	45.8500	18.1200
3754.0000	140.0000	191.0000	32.7000	42.7000	0.0000

5800.0000	200.0000	209.0000	22.1600	54.5000	15.3700
6526.0000	268.0000	223.0000	16.0700	57.9500	0.0000
4718.9478	210.0020	252.2100	25.0000	32.0000	30.0000
8850.0000	279.0000	140.0000	3.7200	9.3700	0.0000
6600.0000	199.0000	232.0000	19.6000	36.2000	0.0000
5161.0229	209.8040	209.8100	26.0000	33.0000	16.0000
4728.0000	260.0000	190.0000	27.7900	57.8400	0.0000
3423.0356	141.9980	141.9900	30.0000	24.0000	41.0000
5475.1746	210.0020	252.2100	25.0000	32.0000	10.0000
6000.0662	249.9980	250.0000	23.0000	50.0000	10.0000
6400.0000	225.0000	140.0000	5.5000	22.6900	0.0000
6400.0000	225.0000	140.0000	5.5000	22.6900	0.0000
3549.9437	249.9980	274.5000	29.7500	26.1500	43.0700
6400.0000	140.0000	232.0000	19.6000	36.2000	0.0000
5652.9909	252.2120	252.2100	25.0000	32.0000	10.0000
5309.9766	245.4260	245.4300	30.2500	26.5700	12.0000
5214.6868	231.0080	231.0000	31.0000	44.0000	27.0000
5923.0511	209.8040	209.8100	26.0000	33.0000	16.0000
7977.0000	268.0000	244.0000	14.0000	60.5500	0.0000
5456.0296	240.6740	240.6800	25.0000	46.0000	18.0000
4650.0000	196.0000	232.0000	32.2000	43.0000	0.0000
5214.6868	231.0080	231.0000	31.0000	44.0000	27.0000
6598.0568	269.9960	270.0000	27.0000	31.0000	0.0000
5497.9455	238.1540	238.1500	24.1200	54.2600	17.8800
7000.0000	300.0000	261.0000	27.0500	21.5400	0.0000
5500.0000	225.0000	215.0000	34.9800	24.6800	0.0000
6300.0042	249.9980	250.0000	23.0000	49.0000	0.0000
6000.0662	249.9980	250.0000	22.0000	50.0000	10.0000
5946.0000	268.0000	218.0000	20.6200	54.6300	0.0000
5472.9990	252.2120	252.2100	25.0000	32.0000	10.0000
6029.9439	252.2120	252.2100	25.0000	32.0000	0.0000
5472.9990	252.2120	252.2100	25.0000	32.0000	10.0000
5427.0221	269.9960	270.0000	27.0000	31.0000	20.0000
7977.0000	268.0000	244.0000	14.0000	60.5500	0.0000
5500.0000	280.0000	230.0000	34.0500	34.6400	0.0000
3219.9828	141.7460	141.7400	33.0000	23.0000	41.0000
6400.0000	140.0000	232.0000	19.6000	36.2000	0.0000
6381.0000	268.0000	221.0000	16.5000	56.6500	0.0000
4774.9324	271.0040	271.0000	27.0000	31.0000	23.0000
5423.9763	210.0020	252.2100	25.0000	32.0000	20.0000
9000.0000	300.0000	140.0000	3.7200	9.3700	0.0000

4500.0000	176.0000	192.0000	14.5500	13.0000	0.0000
6526.0000	268.0000	223.0000	16.0700	57.9500	0.0000
5475.1746	210.0020	252.2100	25.0000	32.0000	10.0000
3423.0356	141.9980	141.9900	30.0000	24.0000	41.0000
5423.9763	210.0020	252.2100	25.0000	32.0000	20.0000
4980.0000	279.0000	140.0000	7.6300	38.6200	0.0000
4800.0000	340.0000	183.6000	26.3700	56.8600	0.0000
5050.0000	280.0000	215.0000	34.9800	24.6800	0.0000
6381.0000	268.0000	221.0000	16.5000	56.6500	0.0000
5472.9990	252.2120	252.2100	25.0000	32.0000	10.0000
5228.7555	121.9100	121.9100	26.0000	33.0000	15.0000
4850.0000	225.0000	140.0000	7.6300	38.6200	0.0000
9000.0000	300.0000	140.0000	3.7200	9.3700	0.0000
5511.5791	132.1160	132.1200	24.0000	54.0000	17.0000
5489.9684	210.0020	252.2100	25.0000	32.0000	15.0000
2680.0073	215.0060	215.0000	24.0000	33.0000	47.0000
5407.0069	209.8040	209.8100	26.0000	33.0000	16.0000
6381.0000	268.0000	221.0000	16.5000	56.6500	0.0000
6000.0000	280.0000	261.0000	33.6300	36.7800	0.0000
5000.0310	294.9620	294.9700	25.0000	37.0000	23.0000
4174.0000	164.0000	193.0000	25.1700	52.6200	0.0000
6400.0000	140.0000	232.0000	19.6000	36.2000	0.0000
3219.9828	141.7460	141.7400	33.0000	23.0000	41.0000
3840.0000	250.0000	195.0000	36.9000	32.7300	0.0000
5456.0296	240.6740	240.6800	25.0000	46.0000	18.0000
5497.9455	238.1540	238.1500	24.1200	54.2600	17.8800
5500.0000	140.0000	232.0000	20.9000	54.7000	0.0000
5467.0000	260.0000	199.0000	25.0600	60.3600	0.0000
5500.0000	140.0000	232.0000	20.9000	54.7000	0.0000
8850.0000	279.0000	140.0000	3.7200	9.3700	0.0000
6700.0000	279.0000	140.0000	5.5000	22.6900	0.0000
4500.0000	176.0000	192.0000	14.5500	13.0000	0.0000
9400.0000	279.0000	140.0000	2.4700	0.0000	0.0000
6017.0000	271.0000	232.0000	20.9000	54.7000	0.0000
3880.0496	302.0000	302.0000	1.0000	42.0000	35.0000
5050.0000	280.0000	215.0000	34.9800	24.6800	0.0000
3880.0496	302.0000	302.0000	1.0000	42.0000	35.0000
3408.9670	254.4080	254.4000	30.3600	23.6400	41.4700
6850.0000	300.0000	140.0000	5.5000	22.6900	0.0000
7000.0000	300.0000	261.0000	27.0500	21.5400	0.0000
5800.0000	200.0000	209.0000	22.1600	54.5000	15.3700

3219.9828	141.7460	141.7400	33.0000	23.0000	41.0000
3549.9437	249.9980	274.5000	29.7500	26.1500	43.0700
3423.0356	141.9980	141.9900	30.0000	24.0000	41.0000
5946.0000	268.0000	218.0000	20.6200	54.6300	0.0000
5475.1746	210.0020	252.2100	25.0000	32.0000	10.0000
6017.0000	271.0000	232.0000	20.9000	54.7000	0.0000
6600.0000	199.0000	232.0000	19.6000	36.2000	0.0000
3840.0000	250.0000	195.0000	36.9000	32.7300	0.0000
5800.0591	208.9940	209.0000	22.0000	55.0000	16.0000
5497.9455	238.1540	238.1500	24.1200	54.2600	17.8800
5800.0000	200.0000	209.0000	22.1600	54.5000	15.3700
5228.7555	121.9100	121.9100	26.0000	33.0000	15.0000
3549.9437	249.9980	274.5000	29.7500	26.1500	43.0700
5652.9909	252.2120	252.2100	25.0000	32.0000	10.0000
5472.9990	252.2120	252.2100	25.0000	32.0000	10.0000
4800.0000	340.0000	183.6000	26.3700	56.8600	0.0000
5753.9371	252.2120	252.2100	25.0000	32.0000	5.0000
5400.4802	240.6740	240.6800	25.0000	46.0000	18.0000
4825.9857	258.0080	258.0000	27.0000	39.0000	24.0000
9400.0000	279.0000	140.0000	2.4700	0.0000	0.0000
2399.9394	215.0060	215.0000	24.0000	33.0000	47.0000
5455.0144	143.7080	143.7000	24.6800	45.8500	18.1200
5511.5791	132.1160	132.1200	24.0000	54.0000	17.0000
4728.0000	260.0000	190.0000	27.7900	57.8400	0.0000
4825.9857	258.0080	258.0000	27.0000	39.0000	24.0000
7000.0000	300.0000	261.0000	27.0500	21.5400	0.0000
5800.0591	208.9940	209.0000	22.0000	55.0000	16.0000
5214.6868	231.0080	231.0000	31.0000	44.0000	27.0000
2750.0605	302.0000	302.0000	1.0000	42.0000	46.0000
4825.9857	258.0080	258.0000	27.0000	39.0000	24.0000
4774.9324	271.0040	271.0000	27.0000	31.0000	23.0000
5489.9684	210.0020	252.2100	25.0000	32.0000	15.0000
2750.0605	302.0000	302.0000	1.0000	42.0000	46.0000

APPENDIX B

MATLAB Code

GMDH_script

```
clc
clf
clear all;%Clears all variables and other classes of data too.
close all;
tic
%to reduce the risk of confusing errors.
%
% Step (1) Reading the input file
% ==============================
% Loads data and prepares it for a neural network.
%ndata= xlsread('all_data.xls');
ndata= xlsread('minimum_miscibility_pressure.xlsx');
%50% of data will be used for training
%25% of data will be used for cross-validation
%25% of data will be used for testing
for i=1:178
    atr(i,:)=ndata(i,:);
end
for i=179:267
    aval(i-178,:)=ndata(i,:);
end
for i=268:length(ndata)
    atest(i-267,:)=ndata(i,:);
end
Ytr=atr(:,1);
Xtr=atr(:,2:6);
Xtst=atest(:,2:6);
Ytst=atest(:,1);
Yv=aval(:,1);
Xv=aval(:,2:6);
[model, time] = gmdhbuild(Xtr, Ytr, 2, 1, 2, 0, 2, 1, 0, Xv, Yv, 0);
gmdheq(model, 3);
[Yqtst] = gmdhpredict(model, Xtst);
[Yqval] = gmdhpredict(model, Xv);
[Yqtr] = gmdhpredict(model, Xtr);
[MSE, RMSE, RRMSE, R2] = gmdhtest(model, Xtst, Ytst);

% Evaluating Relative Error for training set:
%============================================
Et1=(Ytr-Yqtr)./Ytr*100;
[q,z] = size(Et1);
figure

plot(Ytr,Yqtr,'o')
grid off
set(gcf, 'color', 'white')
axis square

title('Predicted Minimum Miscibility Pressure vs. Measured Minimum
Miscibility Pressure');
xlabel('Measured Minimum Miscibility Pressure "psi"');
```

```
ylabel('Predicted Minimum Miscibility Pressure "psi"')
legend('Training set', 'location', 'Northwest')
% Adding Reference Line with 45 degree slope
line([0 ; 10000],[0 ; 10000])
%HINT: Select the y-value based on your data limits

% Evaluating the correlation coefficient for training set:
% ========================================================
Rt1=corrcoef(Yqtr,Ytr);
Rt11=min(Rt1(:,1));
gtext(['correlation coefficient = (' num2str(Rt11) ')']);
hold

% Evaluating Relative Error for validation set:
%==============================================
Ev1=(Yqval-Yv)./Yqval*100;
[m,n] = size(Ev1);
figure

plot(Yv,Yqval,'o')
grid off
set(gcf, 'color', 'white')
axis square

title('Predicted Minimum Miscibility Pressure vs. Measured Minimum
Miscibility Pressure');
xlabel('Measured Minimum Miscibility Pressure "psi"');
ylabel('Predicted Minimum Miscibility Pressure "psi"')
legend('Validation set', 'location', 'Northwest')
% Adding Reference Line with 45 degree slope
line([0 ; 10000],[0 ; 10000])
%HINT: Select the y-value based on your data limits

% Evaluating the correlation coefficient for validation set:
% ========================================================
% for the first target Pressure Drop
Rv1=corrcoef(Yqval,Yv);
Rv11=min(Rv1(:,1));
gtext(['correlation coefficient = (' num2str(Rv11) ')']);
hold

% Evaluating Relative Error for testing set:
%===========================================
% for the first target Pressure Drop
Ett1=(Ytst-Yqtst)./Ytst*100;
[m,n] = size(Ett1);
figure

plot(Ytst,Yqtst,'o')
grid off
set(gcf, 'color', 'white')
axis square

title('Predicted Minimum Miscibility Pressure vs.Measured Minimum
Miscibility Pressure');
xlabel('Measured Minimum Miscibility Pressure "psi"');
ylabel('Predicted Minimum Miscibility Pressure "psi"')
```

```
legend('Testing set', 'location', 'Northwest')
% Addding Reference Line with 45 degree slope
line([0 ; 10000],[0 ; 10000])
%HINT: Select the y-value based on your data limits

% Evaluating the correlation coefficient for testing set:
% ======================================================
Rtt1=corrcoef(Yqtst,Ytst);
Rtt11=min(Rtt1(:,1));
gtext(['correlation coefficient = (' num2str(Rtt11) ')']);
hold

% Plotting the histogram of the errors for training set:
% ======================================================
figure
histfit(Et1,10)
%hist(Et1,10)
h = findobj(gca,'Type','patch');
set(h,'FaceColor','w','EdgeColor','k')
title('Error Distribution for Training Set (Polynomial GMDH Model)');
legend('Training set')
xlabel('Error');
ylabel('Frequency')
set(gcf, 'color', 'white')
hold

% Plotting the histogram of the errors for validation set:
% ======================================================
figure
histfit(Ev1,10)
%hist(Ev1,10)
h = findobj(gca, 'Type', 'patch');
set(h,'FaceColor','w','EdgeColor','k')
title('Error Distribution for Validation Set (Polynomial GMDH Model)');
legend('Validation set')
xlabel('Error');
ylabel('Frequency')
set(gcf, 'color', 'white')
hold

% Plotting the histogram of the errors for testing set:
% ======================================================
figure
histfit(Ett1,10)
%hist(Ett1,10)
h = findobj(gca,'Type','patch');
set(h,'FaceColor','w','EdgeColor','k')
title('Error Distribution for Testing Set (Polynomial GMDH Model)');
legend('Testing set')
xlabel('Error');
ylabel('Frequency')
set(gcf, 'color', 'white')
hold

% Estimating the residuals for training set:
% ==========================================
figure
```

```
Errort1 = Yqtr-Ytr;
plot(Errort1,':ro');
grid off
set(gcf, 'color', 'white')
title('Residual Graph for Training Set (Polynomial GMDH Model)')
legend('Training Set')
xlabel('Data Point No')
ylabel('Errors')
hold

% Estimating the residuals for validation set:
% ============================================
figure
Errorv1 = Yqval-Yv;
plot(Errorv1,':ro');
grid off
set(gcf, 'color', 'white')
title('Residual Graph for Validation Set (Polynomial GMDH Model)')
legend('Validation Set')
xlabel('Data Point No')
ylabel('Errors')
hold

% Estimating the residuals for testing set:
% =========================================
figure
Errortt1 = Yqtst-Ytst;
plot(Errortt1,':ro');
grid off
set(gcf, 'color', 'white')
title('Residual Graph for Testing Set (Polynomial GMDH Model)')
legend('Testing Set')
xlabel('Data Point No')
ylabel('Errors')

% ********************
% STATISTICAL ANALYSIS:
% ********************
% Training set:
% =============
% Determining the Maximum Absolute Percent Relative Error
MaxErrt1 = max(abs(Et1));

% Determining the Minimum Absolute Percent Relative Error
MinErrt1 = min(abs(Et1));

% Evaluating the average error
Etavg1 = 1/q*sum(Et1);

% Evaluating the standard deviation
STDT1 = std(Errort1);

% Evaluating Average Absolute Percent Relative Error
AAPET1 = sum(abs(Et1))/q;

% Evaluating Average Percent Relative Error
APET1 = 1/q*sum(Et1);
```

```
% Evaluating Root Mean Square
RMSET1 = sqrt(sum(abs(Et1).^2)/q);

% Validation set:
% ===============
% Determining the Maximum Absolute Percent Relative Error
MaxErrv1 = max(abs(Ev1));

% Determining the Minimum Absolute Percent Relative Error
MinErrv1 = min(abs(Ev1));

% Evaluating the average error
Evavg1 = 1/m*sum(Ev1);

% Evaluating the standard deviation
STDV1 = std(Errorv1);

% Evaluating Average Absolute Percent Relative Error
AAPEV1 = sum(abs(Ev1))/m;

% Evaluating Average Percent Relative Error
APEV1 = 1/m*sum(Ev1);

% Evaluating Root Mean Square
RMSEV1 = sqrt(sum(abs(Ev1).^2)/m);

% Testing set:
% ============
% Determining the Maximum Absolute Percent Relative Error
MaxErrtt1 = max(abs(Ett1));

% Determining the Minimum Absolute Percent Relative Error
MinErrtt1 = min(abs(Ett1));

% Evaluating the average error
Ettavg1 = 1/m*sum(Ett1);

% Evaluating the standard deviation
STDTT1 = std(Errortt1);

% Evaluating Average Absolute Percent Relative Error
AAPETT1 = sum(abs(Ett1))/m;

% Evaluating Average Percent Relative Error
APETT1 = 1/m*sum(Ett1);

% Evaluating Root Mean Square
RMSETT1 = sqrt(sum(abs(Ett1).^2)/m);


%-------------------------------------------------------------------------
----------
% Simulation: Variation of C7+ Molecular Weight while fixing the other
parameters
% ------------Molecular Weight of C7+ variation-----------------------
----------
```

```
ps1=[linspace(240.6740,240.6740,10); %RESERVOIR TEMPERATURE
[min=121.9100    max=340.0000   mean=121.9100]
linspace(121.9100,302.0000,10);%MOLECULAR WEIGHT OF C7+ [min=121.9100
max=302.0000   mean=219.8372]
linspace(25.0000,25.0000,10);%MOLE FRACTION OF INTERMEDIATES
[min=1.0000   max=36.9000   mean=22.4017]
linspace(46.0000,46.0000,10);%MOLE FRACTION OF C1 [min=0.0000
max=60.5500    mean=37.0900]
linspace(18.0000,18.0000,10)]';%MOLE FRACTION OF C2+ [min=0.0000
max=47.0000   mean=12.2869]

% Now simulate
[Yq_mwc7] = gmdhpredict(model, ps1);
% Plot Figures for Molecular Weight of C7+ variation
figure
px1=plot(ps1(:,2),Yq_mwc7(:,1),'-rs');
set(gca,'YGrid','off','XGrid','off')
set(gca,'FontSize',12,'LineWidth',2);
set(px1,'LineStyle','-.','LineWidth',1.5,'Color','k','MarkerSize',6)
xlabel('Molecular Weight of C7+','FontSize',12)
ylabel('Minimum Miscibility Pressure (psi)', 'fontsize',12)

%-------------------------------------------------------------------
------
% Simulation: Variation of C1 Mole Fraction while fixing the other
parameters
% ------------Mole Fraction of C1 variation--------------------------
------

ps2=[linspace(240.6740,240.6740,10); %RESERVOIR TEMPERATURE
[min=121.9100    max=340.0000   mean=121.9100]
linspace(240.6800,240.6800,10);%MOLECULAR WEIGHT OF C7+ [min=121.9100
max=302.0000   mean=219.8372]
linspace(25.0000,25.0000,10);%MOLE FRACTION OF INTERMEDIATES
[min=1.0000   max=36.9000   mean=22.4017]
linspace(0.0000,60.5500,10);%MOLE FRACTION OF C1 [min=0.0000
max=60.5500    mean=37.0900]
linspace(18.0000,18.0000,10)]';%MOLE FRACTION OF C2+ [min=0.0000
max=47.0000   mean=12.2869]

% Now simulate
[Yq_mfc1] = gmdhpredict(model, ps2);
% Plot Figures for Mole Fraction of C1 variation
figure
px2=plot(ps2(:,4),Yq_mfc1(:,1),'-rs');
set(gca,'YGrid','off','XGrid','off')
set(gca,'FontSize',12,'LineWidth',2);
set(px2,'LineStyle','-.','LineWidth',1.5,'Color','k','MarkerSize',6)
xlabel('Mole Fraction of C1 (%)','FontSize',12)
ylabel('Minimum Miscibility Pressure (psi)', 'fontsize',12)

%-------------------------------------------------------------------
-------
% Simulation: Variation of C2+ Mole Fraction while fixing the other
parameters
```

```
% ------------Mole Fraction of C2+ variation--------------------------
-------

ps3=[linspace(240.6740,240.6740,10); %RESERVOIR TEMPERATURE
[min=121.9100    max=340.0000   mean=121.9100]
linspace(240.6800,240.6800,10);%MOLECULAR WEIGHT OF C7+ [min=121.9100
max=302.0000   mean=219.8372]
linspace(25.0000,25.0000,10);%MOLE FRACTION OF INTERMEDIATES
[min=1.0000   max=36.9000   mean=22.4017]
linspace(46.0000,46.0000,10);%MOLE FRACTION OF C1 [min=0.0000
max=60.5500    mean=37.0900]
linspace(0.0000,47.0000,10)]';%MOLE FRACTION OF C2+ [min=0.0000
max=47.0000   mean=12.2869]

% Now simulate
[Yq_mfc2] = gmdhpredict(model, ps3);
% Plot Figures for Mole Fraction of C2+ variation
figure
px3=plot(ps3(:,5),Yq_mfc2(:,1),'-rs');
set(gca,'YGrid','off','XGrid','off')
set(gca,'FontSize',12,'LineWidth',2);
set(px3,'LineStyle','-.','LineWidth',1.5,'Color','k','MarkerSize',6)
xlabel('Mole Fraction of C2+ (%)','FontSize',12)
ylabel('Minimum Miscibility Pressure (psi)', 'fontsize',12)
```

gmdhbuild.m

```
function [model, time] = gmdhbuild(Xtr, Ytr, maxNumInputs, inputsMore,
...
maxNumNeurons, decNumNeurons, p, critNum, delta, Xv, Yv, verbose)
% GMDHBUILD
% Builds a GMDH-type polynomial neural network using a simple
% layer-by-layer approach
%
% Call
%   [model, time] = gmdhbuild(Xtr, Ytr, maxNumInputs, inputsMore,
maxNumNeurons,
%                   decNumNeurons, p, critNum, delta, Xv, Yv, verbose)
%   [model, time] = gmdhbuild(Xtr, Ytr, maxNumInputs, inputsMore,
maxNumNeurons,
%                   decNumNeurons, p, critNum, delta, Xv, Yv)
%   [model, time] = gmdhbuild(Xtr, Ytr, maxNumInputs, inputsMore,
maxNumNeurons,
%                   decNumNeurons, p, critNum, delta)
%   [model, time] = gmdhbuild(Xtr, Ytr, maxNumInputs, inputsMore,
maxNumNeurons,
%                   decNumNeurons, p, critNum)
%   [model, time] = gmdhbuild(Xtr, Ytr, maxNumInputs, inputsMore,
maxNumNeurons,
%                   decNumNeurons, p)
%   [model, time] = gmdhbuild(Xtr, Ytr, maxNumInputs, inputsMore,
maxNumNeurons,
%                   decNumNeurons)
%   [model, time] = gmdhbuild(Xtr, Ytr, maxNumInputs, inputsMore,
maxNumNeurons)
%   [model, time] = gmdhbuild(Xtr, Ytr, maxNumInputs, inputsMore)
```

```
%   [model, time] = gmdhbuild(Xtr, Ytr, maxNumInputs)
%   [model, time] = gmdhbuild(Xtr, Ytr)
%
% Input
% Xtr, Ytr     : Training data points (Xtr(i,:), Ytr(i)), i = 1,...,n
% maxNumInputs : Maximum number of inputs for individual neurons - if
set
%                to 3, both 2 and 3 inputs will be tried (default = 2)
% inputsMore   : Set to 0 for the neurons to take inputs only from the
%                preceding layer, set to 1 to take inputs also from the
%                original input variables (default = 1)
% maxNumNeurons: Maximal number of neurons in a layer (default = equal
to
%                the number of the original input variables)
% decNumNeurons: In each following layer decrease the number of allowed
%                neurons by decNumNeurons until the number is equal to
1
%                (default = 0)
% p            : Degree of polynomials in neurons (allowed values are 2
and
%                3) (default = 2)
% critNum      : Criterion for evaluation of neurons and for stopping.
%                In each layer only the best neurons (according to the
%                criterion) are retained, and the rest are discarded.
%                (default = 2)
%                0 = use validation data (Xv, Yv)
%                1 = use validation data (Xv, Yv) as well as training
data
%                2 = use Corrected Akaike's Information Criterion
(AICC)
%                3 = use Minimum Description Length (MDL)
%                Note that both choices 0 and 1 correspond to the so
called
%                "regularity criterion".
% delta        : How much lower the criterion value of the network's
new
%                layer must be comparing the the network's preceding
layer
%                (default = 0, which means that new layers will be
added as
%                long as the value gets better (smaller))
% Xv, Yv       : Validation data points (Xv(i,:), Yv(i)), i = 1,...,nv
%                (used when critNum is equal to either 0 or 1)
% verbose      : Set to 0 for no verbose (default = 1)
%
% Output
% model        : GMDH model - a struct with the following elements:
%    numLayers     : Number of layers in the network
%    d             : Number of input variables in the training data set
%    maxNumInputs  : Maximal number of inputs for neurons
%    inputsMore    : See argument "inputsMore"
%    maxNumNeurons : Maximal number of neurons in a layer
%    p             : See argument "p"
%    critNum       : See argument "critNum"
%    layer         : Full information about each layer (number of
neurons,
```

```
%                     indexes of inputs for neurons, matrix of exponents
for
%                     polynomial, polynomial coefficients)
%                     Note that the indexes of inputs are in range
[1..d] if
%                     an input is one of the original input variables,
and
%                     in range [d+1..d+maxNumNeurons] if an input is
taken
%                     from a neuron in the preceding layer.
% time          : Execution time (in seconds)
%
% Please give a reference to the software web page in any publication
% describing research performed using the software e.g., like this:
% Jekabsons G. GMDH-type Polynomial Neural Networks for Matlab, 2010,
% available at http://www.cs.rtu.lv/jekabsons/

% This source code is tested with Matlab version 7.1 (R14SP3).

%
==========================================================================
==
% GMDH-type polynomial neural network
% Version: 1.5
% Date: June 2, 2011
% Author: Gints Jekabsons (gints.jekabsons@rtu.lv)
% URL: http://www.cs.rtu.lv/jekabsons/
%
% Copyright (C) 2009-2011  Gints Jekabsons
%
% This program is free software: you can redistribute it and/or modify
% it under the terms of the GNU General Public License as published by
% the Free Software Foundation, either version 2 of the License, or
% (at your option) any later version.
%
% This program is distributed in the hope that it will be useful,
% but WITHOUT ANY WARRANTY; without even the implied warranty of
% MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
% GNU General Public License for more details.
%
% You should have received a copy of the GNU General Public License
% along with this program. If not, see <http://www.gnu.org/licenses/>.
%
==========================================================================
==

if nargin < 2
    error('Too few input arguments.');
end

[n, d] = size(Xtr);
[ny, dy] = size(Ytr);
if (n < 2) || (d < 2) || (ny ~= n) || (dy ~= 1)
    error('Wrong training data sizes.');
end

if nargin < 3
```

```
    maxNumInputs = 2;
elseif (maxNumInputs ~= 2) && (maxNumInputs ~= 3)
    error('Number of inputs for neurons should be 2 or 3.');
end
if (d < maxNumInputs)
    error('Numbet of input variables in the data is lower than the
number of inputs for individual neurons.');
end
if nargin < 4
    inputsMore = 1;
end
if (nargin < 5) || (maxNumNeurons <= 0)
    maxNumNeurons = d;
end
if maxNumNeurons > d * 2
    error('Too many neurons in a layer. Maximum is two times the number
of input variables.');
end
if maxNumNeurons < 1
    error('Too few neurons in a layer. Minimum is 1.');
end
if (nargin < 6) || (decNumNeurons < 0)
    decNumNeurons = 0;
end
if nargin < 7
    p = 2;
elseif (p ~= 2) && (p ~= 3)
    error('Degree of individual neurons should be 2 or 3.');
end
if nargin < 8
    critNum = 2;
end
if any(critNum == [0,1,2,3]) == 0
    error('Only four values for critNum are available (0,1 - use
validation data; 2 - AICC; 3 - MDL).');
end
if nargin < 9
    delta = 0;
end
if (nargin < 11) && (critNum <= 1)
    error('Evaluating the models in validation data requires validation
data set.');
end
if (nargin >= 11) && (critNum <= 1)
    [nv, dv] = size(Xv);
    [nvy, dvy] = size(Yv);
    if (nv < 1) || (dv ~= d) || (nvy ~= nv) || (dvy ~= 1)
        error('Wrong validation data sizes.');
    end
end
if nargin < 12
    verbose = 1;
end

ws = warning('off');
if verbose ~= 0
    fprintf('Building GMDH-type neural network...\n');
```

```
end
tic;

if p == 2
    numTermsReal = 6 + 4 * (maxNumInputs == 3); %6 or 10 terms
else
    numTermsReal = 10 + 10 * (maxNumInputs == 3); %10 or 20 terms
end

Xtr(:, d+1:d+maxNumNeurons) = zeros(n, maxNumNeurons);
if critNum <= 1
    Xv(:, d+1:d+maxNumNeurons) = zeros(nv, maxNumNeurons);
end

%start the main loop and create layers
model.numLayers = 0;
while 1

    if verbose ~= 0
        fprintf('Building layer #%d...\n', model.numLayers + 1);
    end

    layer(model.numLayers + 1).numNeurons = 0;
    modelsTried = 0;
    layer(model.numLayers + 1).coefs = zeros(maxNumNeurons,
numTermsReal);

    for numInputsTry = maxNumInputs:-1:2

        %create matrix of exponents for polynomials
        if p == 2
            numTerms = 6 + 4 * (numInputsTry == 3); %6 or 10 terms
            if numInputsTry == 2
                r = [0,0;0,1;1,0;1,1;0,2;2,0];
            else
                r =
[0,0,0;0,0,1;0,1,0;1,0,0;0,1,1;1,0,1;1,1,0;0,0,2;0,2,0;2,0,0];
            end
        else
            numTerms = 10 + 10 * (numInputsTry == 3); %10 or 20 terms
            if numInputsTry == 2
                r = [0,0;0,1;1,0;1,1;0,2;2,0;1,2;2,1;0,3;3,0];
            else
                r =
[0,0,0;0,0,1;0,1,0;1,0,0;0,1,1;1,0,1;1,1,0;0,0,2;0,2,0;2,0,0; ...

1,1,1;0,1,2;0,2,1;1,0,2;1,2,0;2,0,1;2,1,0;0,0,3;0,3,0;3,0,0];
            end
        end

        %create matrix of all combinations of inputs for neurons
        if model.numLayers == 0
            combs = nchoosek(1:1:d, numInputsTry);
        else
            if inputsMore == 1
                combs = nchoosek([1:1:d
d+1:1:d+layer(model.numLayers).numNeurons], numInputsTry);
```

```
            else
                combs =
nchoosek(d+1:1:d+layer(model.numLayers).numNeurons, numInputsTry);
            end
        end
        %delete all combinations in which none of the inputs are from
the preceding layer
        if model.numLayers > 0
            i = 1;
            while i <= size(combs,1)
                if all(combs(i,:) <= d)
                    combs(i,:) = [];
                else
                    i = i + 1;
                end
            end
        end

        makeEmpty = 1;

        %try all the combinations of inputs for neurons
        for i = 1 : size(combs,1)

            %create matrix for all polynomial terms
            Vals = ones(n, numTerms);
            if critNum <= 1
                Valsv = ones(nv, numTerms);
            end
            for idx = 2 : numTerms
                bf = r(idx, :);
                t = bf > 0;
                tmp = Xtr(:, combs(i,t)) .^ bf(ones(n, 1), t);
                if critNum <= 1
                    tmpv = Xv(:, combs(i,t)) .^ bf(ones(nv, 1), t);
                end
                if size(tmp, 2) == 1
                    Vals(:, idx) = tmp;
                    if critNum <= 1
                        Valsv(:, idx) = tmpv;
                    end
                else
                    Vals(:, idx) = prod(tmp, 2);
                    if critNum <= 1
                        Valsv(:, idx) = prod(tmpv, 2);
                    end
                end
            end

            %calculate coefficients and evaluate the network
            coefs = (Vals' * Vals) \ (Vals' * Ytr);
            modelsTried = modelsTried + 1;
            if ~isnan(coefs(1))
                predY = Vals * coefs;
                if critNum <= 1
                    predYv = Valsv * coefs;
                    if critNum == 0
                        crit = sqrt(mean((predYv - Yv).^2));
```

```
                    else
                        crit = sqrt(mean([(predYv - Yv).^2; (predY -
Ytr).^2]));
                    end
                else
                    comp = complexity(layer, model.numLayers,
maxNumNeurons, d, combs(i,:)) + size(coefs, 2);
                    if critNum == 2 %AICC
                        if (n-comp-1 > 0)
                            crit = n*log(mean((predY - Ytr).^2)) +
2*comp + 2*comp*(comp+1)/(n-comp-1);
                        else
                            coefs = NaN;
                        end
                    else %MDL
                        crit = n*log(mean((predY - Ytr).^2)) +
comp*log(n);
                    end
                end
            end

            if ~isnan(coefs(1))
                %add the neuron to the layer if
                %1) the layer is not full;
                %2) the new neuron is better than an existing worst
one.
                maxN = maxNumNeurons - model.numLayers * decNumNeurons;
                if maxN < 1, maxN = 1; end;
                if layer(model.numLayers + 1).numNeurons < maxN
                    %when the layer is not yet full
                    if (maxNumInputs == 3) && (numInputsTry == 2)
                        layer(model.numLayers +
1).coefs(layer(model.numLayers + 1).numNeurons+1, :) = [coefs'
zeros(1,4+6*(p == 3))];
                        layer(model.numLayers +
1).inputs(layer(model.numLayers + 1).numNeurons+1, :) = [combs(i, :)
0];
                    else
                        layer(model.numLayers +
1).coefs(layer(model.numLayers + 1).numNeurons+1, :) = coefs;
                        layer(model.numLayers +
1).inputs(layer(model.numLayers + 1).numNeurons+1, :) = combs(i, :);
                    end
                    layer(model.numLayers +
1).comp(layer(model.numLayers + 1).numNeurons+1) = length(coefs);
                    layer(model.numLayers +
1).crit(layer(model.numLayers + 1).numNeurons+1) = crit;
                    layer(model.numLayers +
1).terms(layer(model.numLayers + 1).numNeurons+1).r = r;
                    if makeEmpty == 1
                        Xtr2 = [];
                        if critNum <= 1
                            Xv2 = [];
                        end
                        makeEmpty = 0;
                    end
```

```
                    Xtr2(:, layer(model.numLayers + 1).numNeurons+1) =
predY;
                    if critNum <= 1
                        Xv2(:, layer(model.numLayers + 1).numNeurons+1)
= predYv;
                    end
                    if (layer(model.numLayers + 1).numNeurons == 0) ||
...
                       (layer(model.numLayers + 1).crit(worstOne) <
crit)
                        worstOne = layer(model.numLayers +
1).numNeurons + 1;
                    end
                    layer(model.numLayers + 1).numNeurons =
layer(model.numLayers + 1).numNeurons + 1;
                else
                    %when the layer is already full
                    if (layer(model.numLayers + 1).crit(worstOne) >
crit)
                        if (maxNumInputs == 3) && (numInputsTry == 2)
                            layer(model.numLayers + 1).coefs(worstOne,
:) = [coefs' zeros(1,4+6*(p == 3))];
                            layer(model.numLayers + 1).inputs(worstOne,
:) = [combs(i, :) 0];
                        else
                            layer(model.numLayers + 1).coefs(worstOne,
:) = coefs;
                            layer(model.numLayers + 1).inputs(worstOne,
:) = combs(i, :);
                        end
                        layer(model.numLayers + 1).comp(worstOne) =
length(coefs);
                        layer(model.numLayers + 1).crit(worstOne) =
crit;
                        layer(model.numLayers + 1).terms(worstOne).r =
r;
                        Xtr2(:, worstOne) = predY;
                        if critNum <= 1
                            Xv2(:, worstOne) = predYv;
                        end
                        [dummy, worstOne] = max(layer(model.numLayers +
1).crit);
                    end
                end
            end

        end

    end

    if verbose ~= 0
        fprintf('Neurons tried in this layer: %d\n', modelsTried);
        fprintf('Neurons included in this layer: %d\n',
layer(model.numLayers + 1).numNeurons);
        if critNum <= 1
            fprintf('RMSE in the validation data of the best neuron:
%f\n', min(layer(model.numLayers + 1).crit));
```

```
        else
            fprintf('Criterion value of the best neuron: %f\n',
min(layer(model.numLayers + 1).crit));
        end
    end

    %stop the process if there are too few neurons in the new layer
    if ((inputsMore == 0) && (layer(model.numLayers + 1).numNeurons <
2)) || ...
       ((inputsMore == 1) && (layer(model.numLayers + 1).numNeurons <
1))
        if (layer(model.numLayers + 1).numNeurons > 0)
            model.numLayers = model.numLayers + 1;
        end
        break
    end

    %if the network got "better", continue the process
    if (layer(model.numLayers + 1).numNeurons > 0) && ...
       ((model.numLayers == 0) || ...
        (min(layer(model.numLayers).crit) - min(layer(model.numLayers +
1).crit) > delta) ) %(min(layer(model.numLayers + 1).crit) <
min(layer(model.numLayers).crit)) )
        model.numLayers = model.numLayers + 1;
    else
        if model.numLayers == 0
            warning(ws);
            error('Failed.');
        end
        break
    end

    %copy the output values of this layer's neurons to the training
    %data matrix
    Xtr(:, d+1:d+layer(model.numLayers).numNeurons) = Xtr2;
    if critNum <= 1
        Xv(:, d+1:d+layer(model.numLayers).numNeurons) = Xv2;
    end

end

model.d = d;
model.maxNumInputs = maxNumInputs;
model.inputsMore = inputsMore;
model.maxNumNeurons = maxNumNeurons;
model.p = p;
model.critNum = critNum;

%only the neurons which are actually used (directly or indirectly) to
%compute the output value may stay in the network
[dummy best] = min(layer(model.numLayers).crit);
model.layer(model.numLayers).coefs(1,:) =
layer(model.numLayers).coefs(best,:);
model.layer(model.numLayers).inputs(1,:) =
layer(model.numLayers).inputs(best,:);
model.layer(model.numLayers).terms(1).r =
layer(model.numLayers).terms(best).r;
```

```
model.layer(model.numLayers).numNeurons = 1;
if model.numLayers > 1
    for i = model.numLayers-1:-1:1 %loop through all the layers
        model.layer(i).numNeurons = 0;
        for k = 1 : layer(i).numNeurons %loop through all the neurons
in this layer
            newNum = 0;
            for j = 1 : model.layer(i+1).numNeurons %loop through all
the neurons which will stay in the next layer
                for jj = 1 : maxNumInputs %loop through all the inputs
                    if k == model.layer(i+1).inputs(j,jj) - d
                        if newNum == 0
                            model.layer(i).numNeurons =
model.layer(i).numNeurons + 1;

model.layer(i).coefs(model.layer(i).numNeurons,:) =
layer(i).coefs(k,:);

model.layer(i).inputs(model.layer(i).numNeurons,:) =
layer(i).inputs(k,:);

model.layer(i).terms(model.layer(i).numNeurons).r =
layer(i).terms(k).r;
                            newNum = model.layer(i).numNeurons + d;
                            model.layer(i+1).inputs(j,jj) = newNum;
                        else
                            model.layer(i+1).inputs(j,jj) = newNum;
                        end
                        break
                    end
                end
            end
        end
    end
end

time = toc;
warning(ws);

if verbose ~= 0
    fprintf('Done.\n');
    used = zeros(d,1);
    for i = 1 : model.numLayers
        for j = 1 : d
            if any(any(model.layer(i).inputs == j))
                used(j) = 1;
            end
        end
    end
    fprintf('Number of layers: %d\n', model.numLayers);
    fprintf('Number of used input variables: %d\n', sum(used));
    fprintf('Execution time: %0.2f seconds\n', time);
end

return

%====================  Auxiliary functions  ====================
```

```
function [comp] = complexity(layer, numLayers, maxNumNeurons, d,
connections)
%calculates the complexity of the network given output neuron's
connections
%(it is assumed that the complexity of a network is equal to the number
of
%all polynomial terms in all it's neurons which are actually connected
%(directly or indirectly) to network's output)
comp = 0;
if numLayers == 0
    return
end
c = zeros(numLayers, maxNumNeurons);
for i = 1 : numLayers
    c(i, :) = layer(i).comp(:)';
end
%{
%unvectorized version:
for j = 1 : length(connections)
    if connections(j) > d
        comp = comp + c(numLayers, connections(j) - d);
        c(numLayers, connections(j) - d) = -1;
    end
end
%}
ind = connections > d;
if any(ind)
    comp = comp + sum(c(numLayers, connections(ind) - d));
    c(numLayers, connections(ind) - d) = -1;
end
%{
%unvectorized version:
for i = numLayers-1:-1:1
    for j = 1 : layer(i).numNeurons
        for k = 1 : layer(i+1).numNeurons
            if (c(i+1, k) == -1) && (c(i, j) > -1) && ...
               any(layer(i+1).inputs(k,:) == j + d)
                comp = comp + c(i, j);
                c(i, j) = -1;
            end
        end
    end
end
%}
for i = numLayers-1:-1:1
        for k = 1 : layer(i+1).numNeurons
            if c(i+1, k) == -1
                inp = layer(i+1).inputs(k,:);
                used = inp > d;
                if any(used)
                    ind = inp(used) - d;
                    ind = ind(c(i, ind) > -1);
                    if ~isempty(ind)
                        comp = comp + sum(c(i, ind));
                        c(i, ind) = -1;
                    end
```

```
            end
        end
    end
end
return
```

gmdheq.m

```
function gmdheq(model, precision)
% gmdheq
% Outputs the equations of GMDH model.
%
% Call
%   gmdheq(model, precision)
%   gmdheq(model)
%
% Input
%   model         : GMDH-type model
%   precision     : Number of digits in the model coefficients
%                   (default = 15)

if nargin < 1
    error('Too few input arguments.');
end
if (nargin < 2) || (isempty(precision))
    precision = 15;
end

if model.numLayers > 0
    p = ['%.' num2str(precision) 'g'];
    fprintf('Number of layers: %d\n', model.numLayers);
    for i = 1 : model.numLayers %loop through all the layers
        fprintf('Layer #%d\n', i);
        fprintf('Number of neurons: %d\n', model.layer(i).numNeurons);
        for j = 1 : model.layer(i).numNeurons %loop through all the
neurons in the ith layer
            [terms inputs] = size(model.layer(i).terms(j).r); %number
of terms and inputs
            if (i == model.numLayers)
                str = ['y = ' num2str(model.layer(i).coefs(j,1),p)];
            else
                str = ['x' num2str(j + i*model.d) ' = '
num2str(model.layer(i).coefs(j,1),p)];
            end
            for k = 2 : terms %loop through all the terms
                if model.layer(i).coefs(j,k) >= 0
                    str = [str ' +'];
                else
                    str = [str ' '];
                end
                str = [str num2str(model.layer(i).coefs(j,k),p)];
                for kk = 1 : inputs %loop through all the inputs
                    if (model.layer(i).terms(j).r(k,kk) > 0)
                        for kkk = 1 : model.layer(i).terms(j).r(k,kk)
                            if (model.layer(i).inputs(j,kk) <= model.d)
```

```
                                str = [str '*x'
num2str(model.layer(i).inputs(j,kk))];
                            else
                                str = [str '*x'
num2str(model.layer(i).inputs(j,kk) + (i-2)*model.d)];
                            end
                        end
                    end
                end
            end
            disp(str);
        end
    end
else
    disp('The network has zero layers.');
end

return
```

gmdhpredict.m

```
function Yq = gmdhpredict(model, Xq)
% GMDHPREDICT
% Predicts output values for the given query points Xq using a GMDH
model
%
% Call
%   [Yq] = gmdhpredict(model, Xq)
%
% Input
% model     : GMDH model
% Xq        : Inputs of query data points (Xq(i,:)), i = 1,...,nq
%
% Output
% Yq        : Predicted outputs of query data points (Yq(i)), i =
1,...,nq

% This source code is tested with Matlab version 7.1 (R14SP3).

if nargin < 2
    error('Too few input arguments.');
end
if model.d ~= size(Xq, 2)
    error('The matrix should have the same number of columns as the
matrix with which the network was built.');
end

[n, d] = size(Xq);
Yq = zeros(n, 1);

for q = 1 : n
    for i = 1 : model.numLayers
        if i ~= model.numLayers
            Xq_tmp = zeros(1, model.layer(i).numNeurons);
        end
        for j = 1 : model.layer(i).numNeurons
```

```
            %create matrix for all polynomial terms
            numTerms =  size(model.layer(i).terms(j).r,1);
            Vals = ones(numTerms,1);
            for idx = 2 : numTerms
                bf = model.layer(i).terms(j).r(idx, :);
                t = bf > 0;
                tmp = Xq(q, model.layer(i).inputs(j,t)) .^ bf(1, t);
                if size(tmp, 2) == 1
                    Vals(idx,1) = tmp;
                else
                    Vals(idx,1) = prod(tmp, 2);
                end
            end

            %predict output value
            predY = model.layer(i).coefs(j,1:numTerms) * Vals;
            if i ~= model.numLayers
                %Xq(q, d+j) = predY;
                Xq_tmp(j) = predY;
            else
                Yq(q) = predY;
            end

        end
        if i ~= model.numLayers
            Xq(q, d+1:d+model.layer(i).numNeurons) = Xq_tmp;
        end
    end
end

return
```

gmdhtest.m

```
function [MSE, RMSE, RRMSE, R2] = gmdhtest(model, Xtst, Ytst)
% GMDHTEST
% Tests a GMDH-type network model on a test data set (Xtst, Ytst)
%
% Call
%   [MSE, RMSE, RRMSE, R2] = gmdhtest(model, Xtst, Ytst)
%
% Input
% model     : GMDH model
% Xtst, Ytst: Test data points (Xtst(i,:), Ytst(i)), i = 1,...,ntst
%
% Output
% MSE       : Mean Squared Error
% RMSE      : Root Mean Squared Error
% RRMSE     : Relative Root Mean Squared Error
% R2        : Coefficient of Determination

if nargin < 3
    error('Too few input arguments.');
end
if (size(Xtst, 1) ~= size(Ytst, 1))
```

```
    error('The number of rows in the matrix and the vector should be
equal.');
end
if model.d ~= size(Xtst, 2)
    error('The matrix should have the same number of columns as the
matrix with which the model was built.');
end
MSE = mean((gmdhpredict(model, Xtst) - Ytst) .^ 2);
RMSE = sqrt(MSE);
if size(Ytst, 1) > 1
    RRMSE = RMSE / std(Ytst, 1);
    R2 = 1 - MSE / var(Ytst, 1);
else
    RRMSE = Inf;
    R2 = Inf;
end
return
```